Design and Verification of a Compact Variable Stiffness Actuator With a Very Large Range of Stiffness

Daniel Garces
Marquette University

DESIGN AND VERIFICATION OF A COMPACT VARIABLE STIFFNESS ACTUATOR WITH A VERY LARGE RANGE OF STIFFNESS

by

Daniel R. Garces, B.S.

A Thesis submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science

Milwaukee, Wisconsin

August 2014

ABSTRACT
DESIGN AND VERIFICATION OF A COMPACT VARIABLE STIFFNESS ACTUATOR WITH A VERY LARGE RANGE OF STIFFNESS

Daniel R. Garces, B.S.

Marquette University, 2014

Current conventional robots require high stiffness joints to provide absolute positioning accuracy in free space which also causes problems when operating in constrained space. To circumvent these problems, Variable Stiffness Actuators (VSAs) can be used to vary their stiffness to suit the task being performed. A new VSA was designed to provide a very large range of stiffness in a compact size. The Arched Flexure VSA uses a cantilevered beam acting as the flexure with a variable point of contact. It allows the joint to have continuous variable stiffness, have zero stiffness for a small range of motion, and rapid stiffness change.

Finite element analysis was used to evaluate flexture stiffness. The flexure geometry was optimized for two different objectives. In the first case, the flexture was optimized for maximum stiffness range. This optimization resulted in a stiffness ratio of 1200. In the second case, the flexture was optimized for both maximum stiffness range and constant relative sensitivity. This optimization resulted in a stiffness ratio of 100.

A small proof-of-concept VSA actuator based on the constant relative sensitivity alternative was designed, built, and tested. The VSA provided a stiffness ratio of 55, a little more than half of that expected for the flexure alone. The VSA weighed 1.45 pounds and fits within a 4.5 inch by 2 inch by 5 inch volume. The VSA provides the anticipated free joint range for zero stiffness and provides 360 degrees of rotation. It changes from minimum to maximum stiffness in 0.12 seconds.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Current conventional robots require high stiffness joints to provide absolute positioning accuracy in free space. If a robot of this type were to perform a constrained task, the robot arm would either penetrate the constraint or be stopped by the constraint, damaging themselves or the constraining object in the process. As a consequence, high stiffness joints typically prevent conventional robots from being applied to tasks that require physical interaction with their environment. For example, a robot cannot readily perform assembly tasks, as any offset in the parts will cause assembly failure. Also, a conventional robot cannot interact with people without significant risk of hurting the person.

Robots can circumvent this lost opportunity by using one or more of the following: direct drive, compliant links, redundant actuation, force feedback, end effector compliant tooling, serial elastic actuators, or variable stiffness actuators. These methods introduce some sort of compliance to the system.

Direct drive robots provide a direct connection between the motors and the link they are powering (without gearing). This reduces the impedance (apparent stiffness) of the geartrain and motor combination and allows motors to be back driven by physical constraints. While this allows the robot to perform interaction tasks, it also requires constant motor activation to maintain position to resist gravity forces and significantly reduces robot payload.

Another means of reducing interaction forces is to use robots with links that are compliant. Compliant links decrease the overall stiffness of the robot. This allows the robot to contact constraints, as the links will bend instead of damaging the robot or physical constraint. The major issue with this design is that it reduces the absolute positioning accuracy in free space, e.g., due to gravitational bending of the robot. High speed

applications are also difficult as the links will elastically deform at higher speeds due to inertial loads.

Redundant actuation provides additional degrees of freedom at the robot end effector to improve fine manipulation. The redundant actuation approach typically takes the form of a much smaller set of actuators on the end effector of the robot. This decreases the inertia of the end effector when performing fine manipulation which allows for easier control.

Force feedback consists of a control method that utilizes sensors to compare the actual contact force with the intended contact force. If a discrepancy exists, the robot motion can be altered. Active force control is less responsive due to sensor processing time and can lead to contact instability if the robot and environment are stiff.

End-effector compliant tooling (like a Remote Center of Compliance) reduces end effector stiffness of the robot. Targeted passive stiffness values are used to regulate the contact force while performing the task. In addition, by being located on the end effector they don't have to deal with the inertia of the entire robot, just the tooling and part inertia. They do not reduce the stiffness of the entire robot, making robot impact still dangerous to people and objects. They are also designed for a single specific task. This limits their application as general purpose tools, as they must be designed for each task.

Series Elastic Actuators (SEAs) consist of an actuator connected in series to an elastic element of constant stiffness. This reduced stiffness allows the actuator to better handle impacts and static constraints using both passive compliance and active stiffness control. The passive compliance allows responsive force regulation if misalignment is small. Force regulation for large misalignment is attained using force feedback control. Robots having SEAs, like Baxter by Rethink Robotics, utilize SEAs at all joints to minimize forces for unplanned contact. The SEAs uses a torsional spring between the actuator and the link.

People can dynamically change the stiffness of their joints, and do it regularly without thinking. For example, when hammering a nail, turning a door knob, or taking a step, a person dynamically changes their stiffness to suit the task. When a person hammers

a nail, the person has high stiffness in their wrist and elbow when accelerating the hammer forward. Then, just prior to the hammer impacting the nail, the stiffness is decreased and the shock of the nail impact is imparted to the hammer but not transmitted to the person's arm. A person turning a door knob maintains high stiffness along the rotational axis of the door knob while having low stiffness along all of the other axis. When walking, a person changes the stiffness of their knees and ankles to absorb the impact shock of the step and then provide the energy for the step. Similar concepts apply to kicking a ball, shaking hands, throwing a baseball, and many other everyday activities. This type of actuation will be needed in the next generation of robots capable of performing interaction tasks.

1.1 Related Prior Work

Variable Stiffness Actuators (VSAs) vary their stiffness to suit the task being performed. To achieve better interaction behavior, a VSA can change the stiffness of the joint. This will allow a robot to mirror what a person can do, thereby performing much more complex tasks and vastly improving safety.

An important criterion in evaluating VSA performance is a high stiffness ratio. The stiffness ratio is the maximum stiffness value divided by the minimum stiffness value, and a high stiffness ratio is preferred. Other criterion in evaluating VSA quality are size, range of motion, ability to provide free joints, and rate of stiffness change.

There are four approaches used to achieve variable stiffness actuation: 1) equilibrium controlled stiffness, 2) antagonistic controlled stiffness, 3) mechanically controlled stiffness, and 4) structure controlled stiffness (Albu-Schaffer et al., 2008). An equilibrium controlled VSA uses a spring in series with an actuator and changes the equilibrium point of the spring. An antagonistic controlled VSA uses two similar sized actuators connected to one shaft via elastic elements. A mechanically controlled VSA adjusts the effective stiffness of the system, but uses the full spring length at all times. A structure controlled VSA adjusts the effective stiffness of the system by changing the

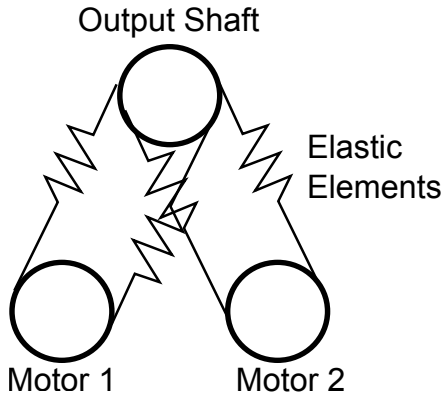


Figure 1.1: The Bidirectional Antagonistic VSA and VSA Cube Schematic Representation. The shaft is connected to both motors via four elastic elements, allowing the motors to work against each other or with each other in either direction.

physical structure of the spring. The performance of previously developed VSAs will be reviewed and compared in the following subsections.

1.1.1 Antagonistic Controlled VSAs

The Bidirectional Antagonistic VSA (Petit et al., 2010) uses a bidirectional antagonistic approach in which both motors are connected to the output shaft by two nonlinear elastic elements each. The four nonlinear elastic elements allow the motors to provide torques in the same direction as well as opposite directions, and enable a larger operating range of torques. This design provides a stiffness ratio of 38. See Figure 1.1 for a schematic representation of the joint. This design does not allow 360 degrees of rotation, instead it has a restricted range of motion of 203 degrees. The Bidirectional Antagonistic VSA changes from minimum to maximum stiffness in 0.014 seconds.

The VSA Cube (Catalano et al., 2011) is a low cost VSA that uses a bidirectional antagonistic approach similar to the Bidirectional Antagonistic VSA. Each of the nonlinear elastic elements are kept in tension by tendons connected to the middle of the elastic elements. The tendons then connect to the output shaft. The tendons connecting to the shaft instead of the springs is the only difference between the Bidirectional Antagonistic VSA

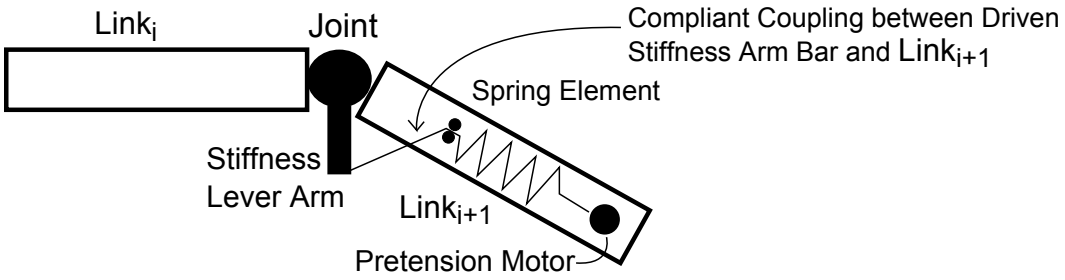


Figure 1.2: The MACCEPA II VSA Schematic Representation. The equilibrium position of the joint is determined by the lever arm, and the second motor changes the pretension in the spring, changing the joint stiffness.

and the VSA Cube. This VSA provides a stiffness ratio of 4.67. See Figure 1.1 for a schematic representation of the joint. It is currently being produced in sample quantities for testing. This design does not allow 360 degrees of rotation, instead it has a restricted range of motion of 120 degrees. This joint is very compact relative to the motors used, but does not provide a large stiffness range, trading range of stiffness for compactness. The VSA Cube was designed to be a stand alone joint system that provides a small stiffness range at low cost for commercial sale. The VSA Cube changes from minimum to maximum stiffness in 0.32 seconds.

1.1.2 Mechanically Controlled VSAs

The MACCEPA II VSA (Vanderborght et al., 2009b) changes joint position by using a lever arm connected by a cable to a spring. As the lever arm changes position, the equilibrium position of the joint changes, creating torque to correct the difference. The joint stiffness is controlled by the pretension in the spring, which changes the equivalent torsional spring seen by the joint. The stiffness of the joint is determined by the pretension in the spring, which is controlled by the second motor retracting cable attached to the end of the spring. This method provides a stiffness ratio of 22. See Figure 1.2 for a schematic representation of the joint. The MACCEPA is currently being used in rehabilitation robots (Cherelle et al., 2010). This method requires a large amount of link space to work, but this is not an issue for

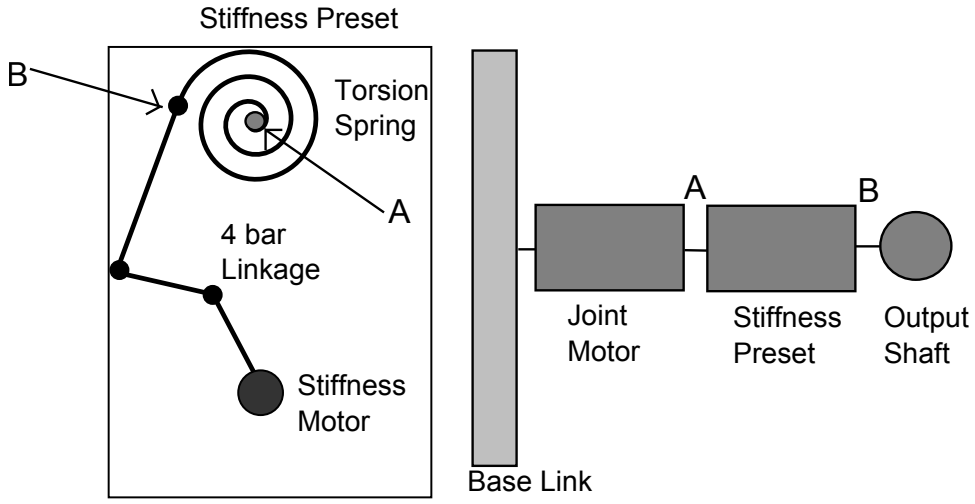


Figure 1.3: The VSA-HD Schematic Representation. The VSA uses a torsion spring connected to a four bar mechanism to change the shaft stiffness.

rehabilitation robots as the link lengths are the person's leg lengths. In addition, this design does not allow 360 degrees of rotation, instead it has a restricted range of motion of 150 degrees. The MACCEPA II changes from minimum to maximum stiffness in 2.6 seconds.

The VSA-HD (Catalano et al., 2010) provides a combination of force feedback and changing stiffness presets to provide a large stiffness ratio of 22000. See Figure 1.3 for a schematic representation of the joint. This design uses a torsion spring connected to a four bar mechanism to change the output shaft stiffness. The stiffness motor changes the stiffness of the torsion spring by applying torques that oppose the position motor's inputs. This design can only provide the maximum range of stiffness values when the output shaft speed is zero, as it requires full motor activation out of both motors. This joint provides 360 degrees of motion, but cannot create a free joint. The VSA-HD provides a compact space, but requires a control system to properly provide the full range of stiffness. The VSA-HD changes from minimum to maximum stiffness in 0.4 seconds.

The DLR Floating Spring Joint (FSJ) (Wolf et al., 2011) was designed to decrease the energy cost of VSAs and provide potential energy storage. The DLR FSJ uses a mechanically controlled approach and changes the stiffness independent of the joint

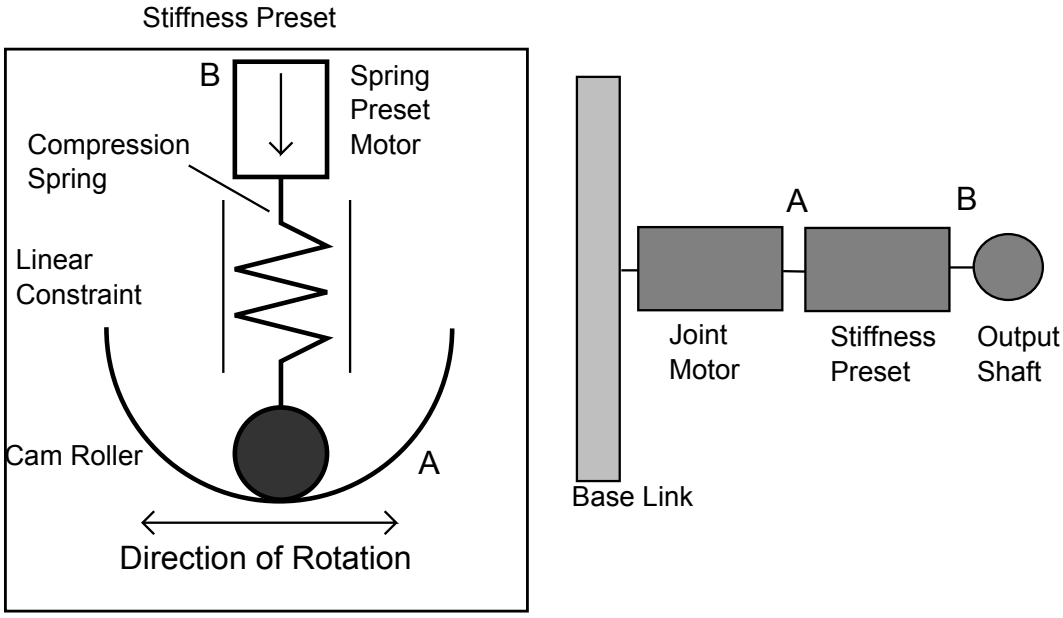


Figure 1.4: The DLR Floating Spring Joint Schematic Representation. The design uses a spring connected to cam disks to change the stiffness.

position. The DLR FSJ tightens a master linear spring connected to cam disks to change the stiffness. As the joint turns, the cam disks apply force against the linear spring, resisting the change in position. The joint provides a stiffness ratio of 16. See Figure 1.4 for a schematic representation of the joint. This joint provides 360 degrees of motion, but cannot create a free joint. The DLR FSJ provides a compact joint that does not require constant motor activation to provide the stiffness change, but does not have a large stiffness ratio. The DLR FSJ changes from minimum to maximum stiffness in 0.33 seconds.

The CompAct-VSA (Tsagarakis et al., 2011) provides a more compact VSA by using an adjustable pivot on a lever arm connected to springs. As the pivot point moves from the force application point to the spring connection point, the stiffness changes due to the changing lever arm lengths. The CompAct-VSA uses the mechanically controlled stiffness method, in which the large motor provides the forces applied to the lever arm and the small motor moves the pivot point changing the stiffness. The stiffness ranges from zero stiffness to rigid. Since the maximum and minimum stiffness is never explicitly tested, this is only a theoretical value. Taking the information from their experimentally determined

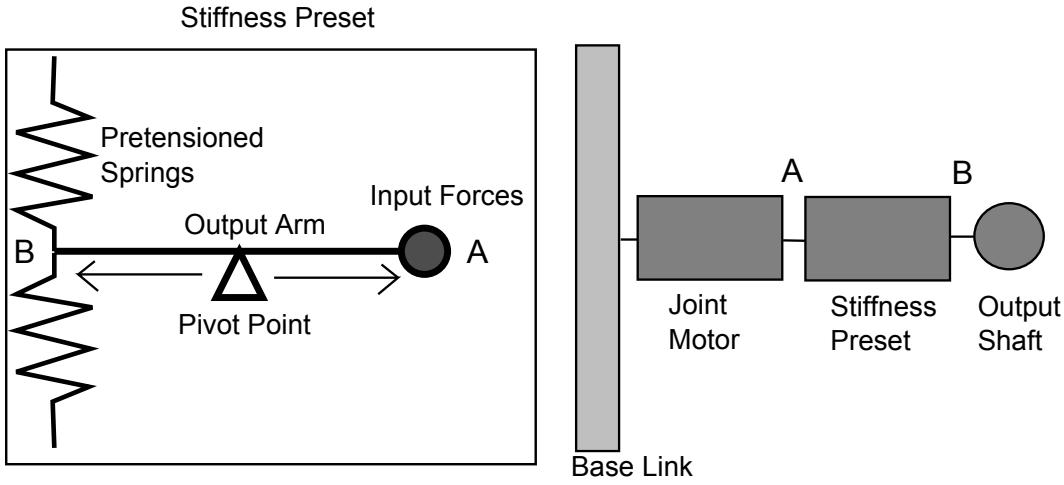


Figure 1.5: The CompAct-VSA and vsaUT-II Schematic Representation. The adjustable pivot provides the stiffness change seen by the output shaft.

stiffnesses, the ratio is at least 35, but likely to be around 50. See Figure 1.5 for a schematic representation of the joint. This design does not allow 360 degrees of rotation, instead it has a restricted range of motion of 120 degrees. The CompAct-VSA provides a compact joint that provides a very large range of stiffness values but does not provide a full revolute joint. The joint also has very nonlinear changes to stiffness values, such that the high stiffness changes occur with very small motor position changes. The CompAct-VSA changes from minimum to maximum stiffness in 0.1 seconds.

The vsaUT-II (Groothuis et al., 2012) uses a lever arm of changing length to connect to internal springs. The vsaUT-II changes the pivot point while keeping the force application point and spring location constant. This is similar to the CompAct-VSA, except this joint uses torsional springs instead of compression springs. The stiffness ranges from zero stiffness to rigid. Since the maximum and minimum stiffness are never explicitly tested, this is only a theoretical value. Taking the information from their experimentally determined stiffnesses, the ratio is at least 50, but likely to be around 70. See Figure 1.5 for a schematic representation of the joint. Note that the vsaUT-II and the CompAct-VSA use the same schematic representation as the only differences are the use of torsional springs vs compression springs, and how those springs are applied to the end of the intermediate link.

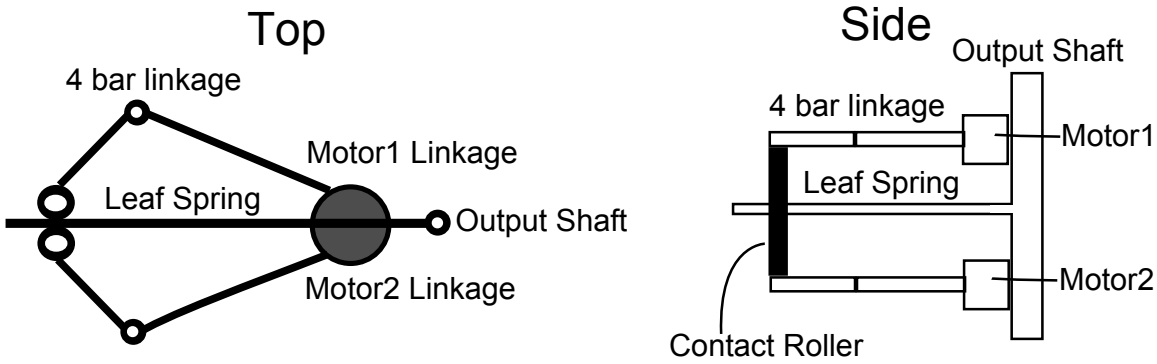


Figure 1.6: The VSJ-Leaf Springs Schematic Representation. The output shaft is connected to both motors through a four bar mechanism contacting leaf springs.

This design does not allow 360 degrees of rotation, instead it has a restricted range of motion of 120 degrees. This joint is not very compact compared to the size of the motors used. The vsaUT-II provides a very large range of stiffness values, but does not provide a full revolute joint or a very compact joint. The vsaUT-II changes from minimum to maximum stiffness in 0.5 seconds.

1.1.3 Structure Controlled VSAs

The VSJ-Leaf Springs (Choi et al., 2011) uses a structure controlled setup with leaf springs as the elastic element. The VSJ-Leaf Springs connects two actuators in parallel via leaf springs, using a four bar mechanism to connect the actuators and the leaf spring. The leaf springs have one free end and one end connected to the output shaft. The design uses four leaf springs to provide greater stiffness values without increasing the footprint of the design. The VSJ-Leaf Springs provides a stiffness ratio of 15. See Figure 1.6 for a schematic representation of the joint. This joint provides 360 degrees of motion, but cannot create a free joint. The VSJ-Leaf Springs is not compact, requiring a large volume relative to the motors used. The VSJ-Leaf Springs changes from minimum to maximum stiffness in 0.2 seconds.

Table 1.1: Summary of the VSA Options and Their Abilities

Name	**Constant Power Needed**	**Estimated Stiffness Ratio**	**Free Joint**	**Range of Motion**	**Force Feedback Required**	**Type**	**Full Range Variation Time**
Bidirectional Antagonistic VSA	Yes	37.5	No	203°	No	Antagonistic	0.014s
VSA Cube	Yes	4.637	No	120°	No	Antagonistic	0.32s
MACCEPA II	Yes	22	No	150°	No	Mechanical	2.6s
VSA-HD	No	22000	No	∞	Yes	Mechanical	0.4s
DLR FSJ	No	15.75	No	180°	No	Mechanical	0.33s
CompAct-VSA	No	50	Yes	120°	No	Mechanical	0.1s
vsaUT-II	No	70	Yes	120°	No	Mechanical	0.5s
VSJ-Leaf Springs	No	14.467	No	∞	No	Structure	0.2s

1.1.4 Summary of Prior VSA Designs

Table 1.1 summarizes the performance of the reviewed VSAs with respect to the stiffness ratio, constant power needs, free joint availability, range of motion, force feedback requirement, and type of VSA. The stiffness ratio indicates the maximum stiffness divided by the minimum stiffness, where a larger value is better. Constant power shows if the design requires constant power to the motors to maintain the stiffness ratio, or if the power is only required to change the stiffness value, where non constant power is preferred. The free joint shows if the design can provide a free joint, which allows the VSA to be used in more applications. The range of motion shows the range of motion the joint can support, where greater ranges are preferred. The force feedback indicates if the joint is passive or active and passive, with passive preferred. The type indicates what method of stiffness control the VSA uses.

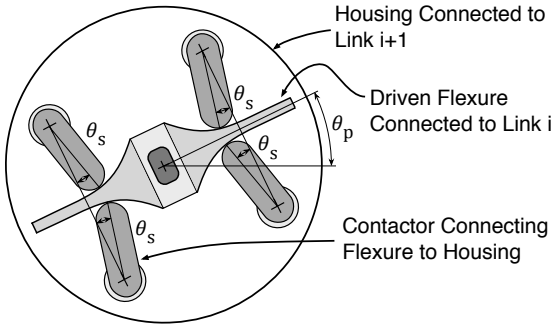


Figure 1.7: The Arched Flexure VSA Schematic Representation. The changing contact points with the flexure provides the stiffness changes.

1.2 Approach

The Arched Flexure Variable Stiffness Actuator (Arched Flexure VSA) described in this thesis provides a very large range of stiffness in a compact size. The Arched Flexure VSA uses a structure controlled stiffness method.

The Arched Flexure VSA changes the physical connection between the flexure (the spring) and the remainder of the VSA to change the stiffness. The VSJ-Leaf Springs operates using a very similar technique. Instead of changing just the length like the VSJ-Leaf Springs, the Arched Flexure VSA also changes the direction the force is applied and the area moment of inertia. This provides greater stiffness change than the VSJ-Leaf Springs, while still providing 360 degrees of rotation in a compact design.

The length, area moment of inertia, and force application angle all change when the contact bar between the flexure and the next link rotates. The key to the large changes in stiffness values comes from all of the values changing to increase the joint stiffness as the contact point moves closer to the flexure center. The VSA allows the joint to have zero stiffness as well. By changing the contact point on the flexure, the joint can change stiffness rapidly. See Figure 1.7 for a schematic representation of the joint. The VSA is capable of providing a wide range of operating stiffnesses for the actuator. This VSA provides 360 degrees of motion. It also can become a free joint for limited ranges. The design only requires small motor power to change stiffness. The VSA should provide a free joint. The

Table 1.2: Arched Flexure VSA Specifications

Metric	**Ideal**	**Marginal**
Stiffness Ratio	$\geq$ 1000	$\geq$ 100
Range of Motion	∞	180°
Full Range Variation Time	0.1s	0.2s
Size	$\leq$ 3in x 3in x 3in	$\leq$ 5in x 5in x 5in
Prototype Cost	$\leq$ $ 5000	$\leq$ $ 10000
Weight	$\leq$ 2lb	$\leq$ 5lb
Max Load	100Nm	50Nm

free joint increases the possible tasks the VSA can perform, and allows it to match current VSA options. If the VSA does not provide a free joint, the VSA will still be fully functional, it just will have less applications that it can be applied to. Finally, the VSA should not require constant power to attain a specific stiffness value, which increases the number of robots the VSA can be applied to.

Table 1.2 summarizes target specifications, including both marginally acceptable and ideal values. These values were selected so that the Arched Flexure VSA will meet, or exceed, current VSA performance. First, it should provide a stiffness ratio greater than 1000. This will provide the best stiffness ratio out of any of the current VSA options. The marginal stiffness ratio of 100 will be better than all of the VSA options, except for the VSA-HD which achieves performance using active control.

Second, it should provide 360 degrees of rotation, which allows the VSA design to be put on current robots. The marginal value of 180 degrees of rotation will still be better than the majority of the VSA options.

Third, it should change from minimum to maximum stiffness as quickly as possible, preferably under 0.1 seconds. Changing stiffness quickly allows the VSA to operate when a sudden change in stiffness is needed, and changing under 0.1 seconds will make it one of the best VSA options in that regard. The marginal value of 0.2 seconds to change stiffness will still beat the majority of the VSA options, it just won't be as effective as the ideal rate.

Fourth, the VSA should be as compact as possible. While there isn't a specific size constraint on the design, the VSA should try to be as compact as possible while still meeting the previous specifications. The more compact the VSA is the more places it can be used, by decreasing the inertia of the VSA.

The VSA should cost less than 5000 dollars. This will allow a commercial Arched Flexure VSA to cost much less than the majority of the alternative VSAs, and allowing it to compete with the VSA Cube design on a cost basis.

Finally, the weight of the VSA should be as low as possible, allowing it to be mounted to more robot designs.

1.3 Overview

The Arched Flexure VSA has several key systems for which the detailed design is described in the remainder of this document. The flexure consists of a nonlinear cantilever beam with dynamic contact points. This allows changes in length, area, and direction of forces to all provide a net benefit to stiffness. In Chapter 2 the flexure is optimized to provide the largest stiffness ratio possible. It is also optimized for maximum stiffness ratio with constant relative sensitivity. The optimization involves the objective function, the constraints, and the method of analysis. Chapter 3 covers the design of the stiffness selection system, the joint actuation system, and the overall joint structure. These systems are all designed around the flexure optimized in Chapter 2 for maximum stiffness ratio with constant relative sensitivity. The manufacture and assembly of the VSA design are also covered in Chapter 3. Chapter 4 compares the VSA experimental results to the theoretical results. Chapter 5 summarizes the benefits of the overall design, and provides recommendations for future work.

CHAPTER 2

DESIGN OF THE FLEXURE

The Arched Flexure VSA provides a very large range of stiffness with constant relative sensitivity, and can provide a free joint. The VSA uses a structure controlled stiffness method, with the elastic behavior provided by the flexure. The flexure consists of a cantilever beam of varying cross section. Contact at different locations along it's length allows the VSA to change it's stiffness value. This chapter presents the design of the flexure that was optimized for use in a VSA.

The flexure design has a great deal of importance to the overall VSA design. If the flexure doesn't provide a large stiffness ratio then the VSA will not provide a large stiffness ratio. The flexure design process has two main components, the geometry selection and the material selection. The geometry selection covers the selection of the flexure shape, impacting the compactness, stiffness range, and functionality of the flexure. The material selection covers the selection of the material used for the flexure, impacting the maximum stiffness, minimum stiffness, and angular deflection. Flexure geometry selection can be made separately from material selection because the flexure should only operate in the linear elastic region.

First, the normalized independent variables used in defining flexure geometry are presented. Then, the objective function and constraints are explained. Next, the optimization for maximum stiffness ratio only and for maximum stiffness with constant relative sensitivity are described. Then, the analysis of the results and discussion of their implications is presented. Then, the free joint range and whisker analysis is described. Finally, the flexure material selection is presented.

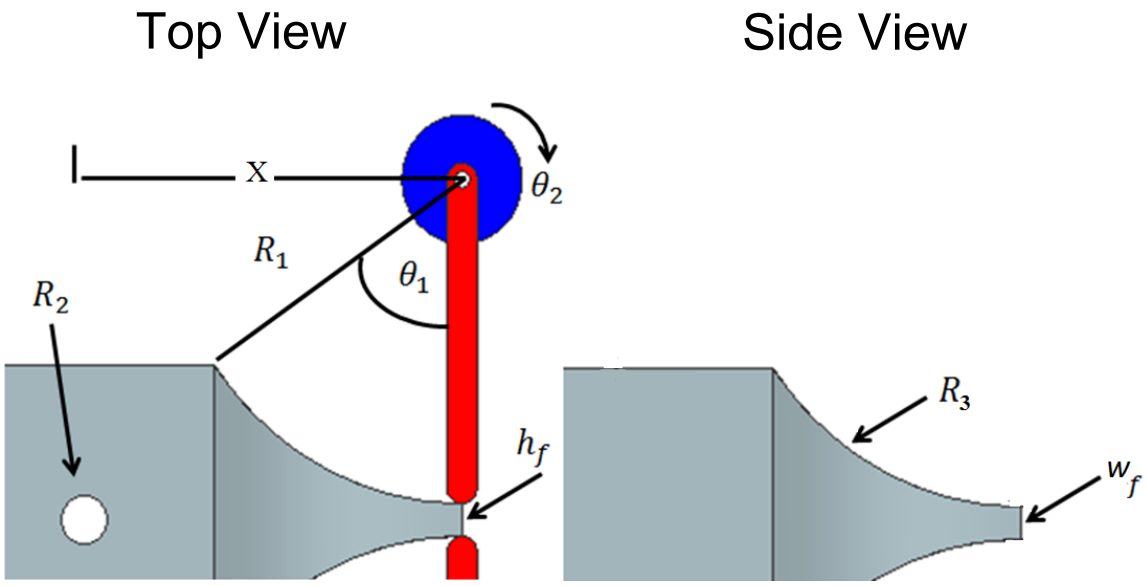


Figure 2.1: Schematic Representation of the Optimization Variables

2.1 Geometry Selection

Parametric design takes a given shape and breaks it into a set of independent variables that are changed independently to modify the flexure shape and performance. They fully define the flexure shape with no additional variables required. Figure 2.1 is a schematic representation of the independent variables. The independent variables are as follows:

- $X \equiv$ X Location of Center of Curvature from Mounting Hole
- $R_1 \equiv$ Radius of Flexure in Plane Parallel to Contact Selection Bars
- $R_2 \equiv$ Radius of the Center Hole
- $R_3 \equiv$ Radius of Flexure in Plane Perpendicular to Contact Selection Bars
- $\theta_1 \equiv$ Maximum Angle of the Curved Portion of Flexure
- $\theta_2 \equiv$ Maximum Force Application Angle
- $h_f \equiv$ Height of Flexure at Distal End

- $w_f \equiv$ Width of Flexure at Distal End

The parameters illustrated in Figure 2.1 are X, R_1, R_2, R_3, θ_1, θ_2, h_f, w_f. These variables are then all normalized to R_2, creating a dimensionless set of independent variables that can then be used for design. The radius of the center hole, R_2, was selected as the normalizing variable to allow the input shaft size from the motor to set the design size. These seven parameters fully define the flexure shape, with θ_2 defining maximum force application angle (θ_2 must be used as an independent variable, as the maximum force application location can change independently, instead of requiring it to be a set angle away from θ_1). The x-location for the center of curvature for R_3 is assumed to be the same as R_1. Using these independent variables, the shape of the flexure is uniquely described.

The analysis of the flexure cannot be performed using standard beam theory, as an assumption necessary for beam theory is violated. Specifically, there is significant change in the area moment of inertia of the beam. Therefore, Finite Element Analysis (FEA) was used to determine its stiffness under different load conditions. FEA takes complex shapes and breaks those shapes into a large number of simple shapes that, when connected together, closely approximate the complex shape being analyzed. In essence, it will break up the flexure into a series of connected cubes and tetrahedrons. This allows the complex flexure shape to be analyzed for its stiffness given different force application angles and locations. The FEA program used for the analysis, ANSYS®, has its own programming language, with a text file input. The full flexure shape is generated in the code provided in Appendix A.

2.1.1 Geometry Optimization

Optimization is used to choose the best flexure shape. An optimization is defined by its objective function and its constraints. The optimization objective function provides a quantifiable means of evaluating design performance. By convention, the objective function is minimized. The constraints place limits on design considerations. The standard representation of an optimization is given by:

$$\min(f(x)) \tag{2.1}$$

$$\text{such that: } g(x) \leq 0 \tag{2.2}$$

$$h(x) = 0 \tag{2.3}$$

where x is the set of design variables.

Objective Function

For the flexure, the selected objective function has a stiffness ratio component and a relative sensitivity component. The objective function was run for two distinct weighting cases. The first case involves only the stiffness ratio component, and the second case involves both the stiffness ratio component and the relative sensitivity component. Equation 2.4 is the general objective function used:

$$f(x) = \left(B * D - \log\left(\frac{k_{\max}}{k_{\min}}\right)\right) \tag{2.4}$$

where B is the weighting factor, D is the least squares fit error, and k is the joint stiffness.

The joint stiffness, k, is calculated from the force-deflection information obtained from the FEA. The FEA outputs node displacement information for all nodes. The node displacement information for the nodes along the force application line are averaged, and then converted to an angular displacement.

$$\delta_\theta = \arctan\left(\frac{\delta_y}{l - \delta_l}\right) \tag{2.5}$$

where δ_θ is the angular displacement, δ_y is the change in height, l is the length of the flexure from the axis of rotation to the contact location, δ_l is the change in length. The applied torque, T, is then divided by the angular displacement to get the stiffness for the force

application location, as seen in Equation 2.6.

$$k = \frac{T}{\delta_\theta} \tag{2.6}$$

This is then repeated for all force application locations.

The B term is the weighting term, giving the two different portions of the optimization different emphasis. The relative sensitivity criteria, D, is the least squares fit error of an exponential function to the stiffness data. The relative sensitivity term provides the least squares fit error between the stiffness data produced by ANSYS® and the exponential function that it is fit to. Fitting the curve to an exponential curve allows for a rapid change in stiffness while still keeping the change in stiffness with respect to the current stiffness location constant ($\frac{\delta_k}{k} = \text{Constant}$). This keeps the uncertainty in the stiffness value of the VSA as a function of θ, making it predictable. The stiffness ratio component, $\frac{k_{\max}}{k_{\min}}$, takes the maximum stiffness and compares it to the minimum stiffness, attempting to find the largest difference between the two.

Constraints

Constraints are applied to the optimization to restrict the set of possible solutions. These constraints address physically imposed limits as well as ANSYS® imposed limits. Each constraint applies to one or more limits, and all variables are normalized to R_2. The constraints are:

$$-R_3 - \frac{w_f}{2} + R_3 \cos\theta_1 + 1.5 \leq 0 \tag{2.7}$$

$$-R_1 - \frac{h_f}{2} + R_1 \cos\theta_1 + 1.5 \leq 0 \tag{2.8}$$

$$-X + 1.5 + R_3 \sin\theta_1 \leq 0 \tag{2.9}$$

$$-X + 1.5 + R_1 \sin\theta_1 \leq 0 \tag{2.10}$$

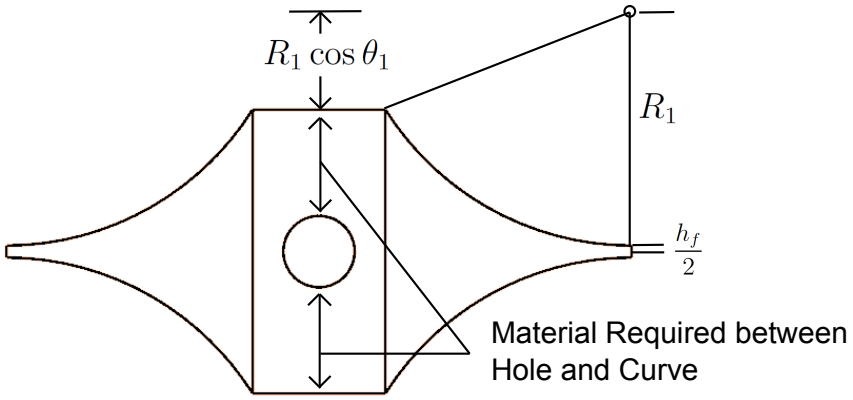


Figure 2.2: Schematic Representation of Constraint 2.8

$$-\theta_1 + \theta_2 \leq \frac{-\pi}{18} \tag{2.11}$$

$$-X + R_1 \leq -0.6667 \tag{2.12}$$

$$-X + R_3 \leq -0.6667 \tag{2.13}$$

$$-\tan\left(\frac{X}{h_f/2 + R_1}\right) + \theta_2 \leq -0.05 \tag{2.14}$$

Constraints 2.7 and 2.8 ensure the shaft mounting hole will not be too large relative to the entire flexure shape, meaning that not enough material would remain to adequately mount the shaft. If these constraints were not included, the optimization could still be run, but the resultant shape would lack structural integrity. Figure 2.2 illustrates the variables used in Constraint 2.8. Constraints 2.9 and 2.10 ensure that the shaft mounting hole is not larger than the flat area of the flexure, keeping the model created in ANSYS® valid. If these constraints are not included, ANSYS® will attempt to create the shape and error out. This is caused by the procedural code ANSYS® uses to create the flexure geometry. See Figure 2.3 for a schematic representation of Constraint 2.10. Constraint 2.11 ensures a gap between the maximum force application angle and the maximum possible angle. Without adequate spacing the contact points could potentially slide off of the flexure. Constraints 2.12 and 2.13 prevent the model generated in ANSYS® from failing, by keeping the procedurally generated shape valid. This is due to the programmed method in which ANSYS® was told

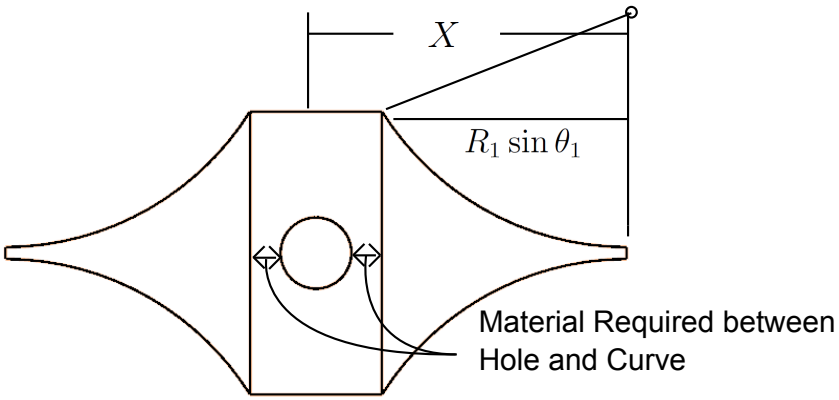


Figure 2.3: Schematic Representation of Constraint 2.10

to generate the shape. The constraints ensure that a portion of the flexure shape can always be created. Constraint 2.14 provides singularity avoidance, ensuring that the flexure and contact selection system avoid the mechanical singularity.

Optimization Methods

There are two main types of optimization, genetic optimization and gradient search optimization. Gradient search optimization evaluates an individual point in the design space and searches the gradient for where to check next. In essence, it takes the point and checks the slope in all directions around it, and then takes a step toward the slope that is changing beneficially. If the point chosen has all positive gradients locally, then the search is concluded and the optimization results in that point as the locally optimal value. This means that the search method is highly influenced by the nature of the objective function. If the objective function has a large number of local minima, then the search yields a local minimum that is unlikely to be the global minimum. To circumvent this, a large number of optimizations can be performed, each with different starting points.

Genetic optimization creates a large population of individuals within the design space, and evaluates the value of the objective function for each individual. It then selects the best individuals of each population (selection) and combines them to create new

individuals (crossover). Periodically it mutates individuals in the population to create significantly different individuals (mutation). Individuals in this generation are then evaluated. The process of selection, crossover, and mutation is repeated until some termination criteria are met, such as the maximum number of generations or the weighted change in the best objective function values. These termination criteria can be modified to suit the optimization problem. The genetic optimization is not guaranteed to find the global minimum, instead it finds an individual that may be close to the minimum. To establish confidence in the result obtained, several populations can be used with a wide variety of initial individuals. If multiple optimizations yield the same or nearly the same individual, it is more likely that the individual is a global minimum.

The genetic optimization routine was selected for this problem for several reasons. First, a large number of local minima exist due to the highly nonlinear space. This means that a gradient search technique would need a large number of seed points to attempt to find the global minimum value, instead of just a local minimum. The genetic optimization bypasses this by providing a large number of individuals in the population. While it is not guaranteed to find the global minimum, it will likely get closer to the global minimum in less time. Second, there is a large space available due to the constraints used. The variable limits constraints were deliberately kept as wide as possible to ensure that the best flexure shapes would be captured. The wide variable range combined with the highly nonlinear space would necessitate a large number of initial seed points for the gradient search technique. Third, the run times of around 7 minutes per call to the optimization for the genetic optimization was 40 times shorter than the gradient search optimization individual run times.

Upper and Lower Bounds

Genetic optimizations require upper and lower bounds on the normalized design variables. These bounds limit the space searched by the optimization. Values were set on the basis of

relative size considerations, and lower bounds on manufacturability. The upper and lower bounds were as follows:

$$\frac{22.5\pi}{180} \leq \theta_1 \leq \frac{85\pi}{180} \tag{2.15}$$

$$6.67 \leq X \leq 33 \tag{2.16}$$

$$0.33 \leq h_f \leq 1.67 \tag{2.17}$$

$$\frac{15\pi}{180} \leq \theta_2 \leq \frac{45\pi}{180} \tag{2.18}$$

$$5 \leq R_1 \leq 13.33 \tag{2.19}$$

$$0.33 \leq w_f \leq 1.67 \tag{2.20}$$

$$5 \leq R_3 \leq 13.33 \tag{2.21}$$

2.1.2 Objective Function Calculation

To refine the optimization, two different FEA models were used optimization runs performed in series. The first model had a coarser grid refinement to reduce computing time. The optimized geometry from the first run was then input as the starting locations for the second optimization. The second optimization used the highest level of grid refinement allowed when using ANSYS® in batch mode.

The first optimization using fewer nodes in the FEA allowed the optimization to be completed in around three minutes. The goal of this initial optimization was to further refine the design space, and realize what constraints, if any, were being pushed up against. The resulting variable values from this optimization can then be input into the second optimization, allowing the individual to be an initial seed point, greatly decreasing overall run time.

To perform the two optimization runs, MatLab® was interfaced with ANSYS®, as MatLab® provides a good optimization toolbox, and ANSYS® provides a good FEA

toolbox. The full code and the method of linking them is provided in Appendix A. After a best case shape was determined, it was then analyzed with a higher level of mesh refinement than possible in batch mode, to confirm that the results from the optimization mesh matched the high mesh refinement result.

2.2 Results

Two sets of results were obtained using the optimization methods described above. The two sets were obtained using different weightings for the objective function. The first optimization looked for maximum stiffness ratio only. The second optimization looked for maximum stiffness ratio as well as constant relative sensitivity.

The first optimization was run with the B weight in Equation 2.4 equal to zero. This meant the optimization was performed involving just the stiffness portion of the objective function, such that it became:

$$f(x) = -\ln\left(\frac{k_{max}}{k_{min}}\right) \tag{2.22}$$

The joint stiffness vs contact selection angle for the maximum stiffness range optimization are shown in Figure 2.4 for the three levels of grid refinement investigated. Figure 2.4 shows that the lower stiffness values all match, whereas at the higher stiffness values they do not match. The calculated stiffness values decrease as the mesh refinement level increases.

See Table 2.1 for the design parameters for the flexure shape optimized for maximum stiffness range. This indicates that the flexure optimization is pushing against the constraint on h_f. This constraint cannot be reduced on the lower bounds, as it is a limit on manufacturability. If the lower bounds were reduced, there is no guarantee that the part can be manufactured. The decreasing pattern at high stiffness levels seen in Figure 2.4 calls the reliability of the FEA into question, as the stiffness doesn't match for all refinement levels.

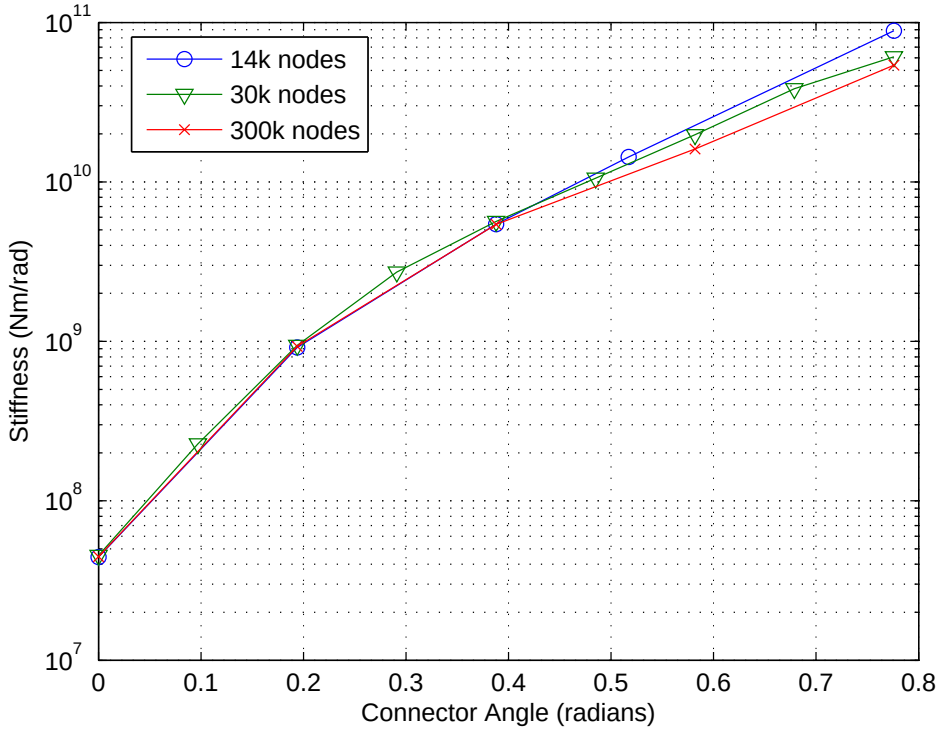


Figure 2.4: Stiffness Values for the Flexure Shape Optimized for Maximum Stiffness Range

Table 2.1: Flexure Design Parameters for the Flexure Shape Optimized for Maximum Stiffness Range

Variable	**Value**
X	8.7333
R_1	8.000
R_2	1
R_3	8.1433
h_f	0.3334
w_f	0.3613
θ_1	1.0077
θ_2	0.7759

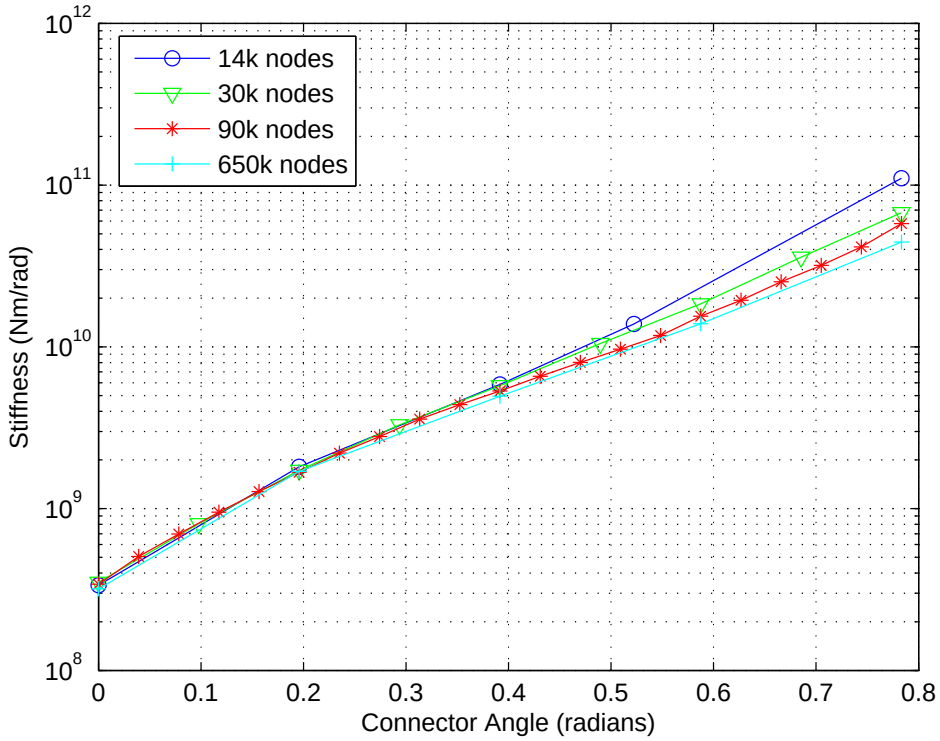


Figure 2.5: Stiffness Values for the Flexure Shape Optimized for Maximum Stiffness Range and Constant Relative Sensitivity

The stiffness ratio provided is 1200. This stiffness ratio exceeds the ideal stiffness ratio specification for the VSA listed in Table 1.2.

For the second optimization B was selected to a value ($B = 0.25$) that places equal emphasis on the two portions of the objective function. This value was determined by numerical experimentation. The joint stiffness vs contact selection angle for maximum stiffness range and relative sensitivity optimization are shown in Figure 2.5. Similar to the previous optimization, Figure 2.5 shows that the lower stiffness values all match at the four levels of mesh refinement, while the higher stiffness values do not match as well. The calculated stiffness values decrease as the mesh refinement level increases. See Table 2.2 for the design parameters for the flexure shape optimized for maximum stiffness range and constant relative sensitivity. The optimization for both maximum stiffness range and relative

Table 2.2: Flexure Design Parameters for the Flexure Shape Optimized for Maximum Stiffness Range and Constant Relative Sensitivity

Variable	**Value**
X	7.7347
R_1	7.2400
R_2	1
R_3	6.6253
h_f	1.0333
w_f	0.5307
θ_1	0.9608
θ_2	0.7834

sensitivity provided a ratio of 100. This stiffness ratio exceeds the ideal stiffness ratio specification from Chapter 1.

The above results are consistent, even though the maximum stiffness decreases as the number of nodes increases. This is due to how ANSYS® handles mesh creation when combined with the approach taken to evaluate stiffness. The low stiffness application points all have meshes that match each other. The high stiffness application meshes only match the low stiffness element density at large mesh refinement levels. See Appendix A for more support and a full explanation of ANSYS® mesh creation and evaluation method.

The optimization results from the maximum stiffness range only optimization can be compared to the optimization results from the relative sensitivity with stiffness optimization. See Figure 2.6 for the CAD models of the two flexures. The CAD models show the differences between the two flexures, and how they vary. Note that the stiffness ratio optimization resulted in a flexure with greater change in area moment of inertia than the relative sensitivity optimization.

Free Joint Range

The flexure has whiskers on the ends of the flexure that extend the flexure past the point where all four contactors can simultaneously be in contact with the flexure. See Figure 2.7 to show the whisker and free joint angle. These whiskers allow the VSA to recapture the

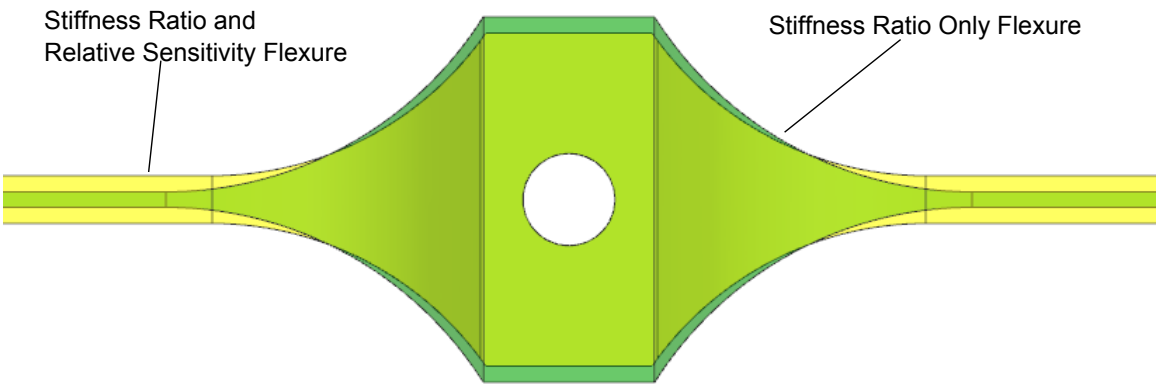


Figure 2.6: Flexure CAD Models for the Two Optimized Flexures

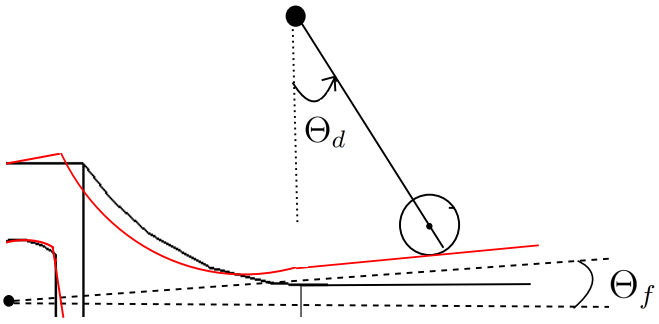


Figure 2.7: Free Joint Angle (Θ_f) with Flexure at Initial Position and Maximum Free Joint Position

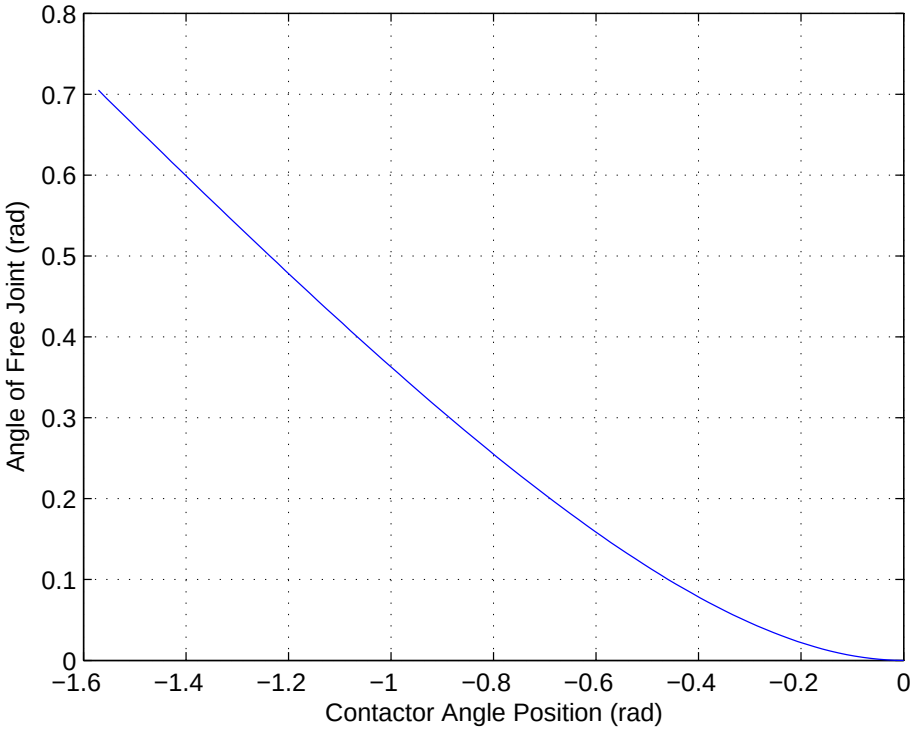


Figure 2.8: Free Joint Angle vs Contactor Angle

flexure after a free joint is provided. The full geometric definition of the free joint angle is described in Appendix A. The calculation for Θ_f assumes that there is no deflection of the flexure, and that it is at it's maximum possible free joint for that given initial angle. Figure 2.8 shows the Free Joint Angle compared to the Contactor Angle.

2.2.1 Flexure Material

To select flexure material the modulus of elasticity, yield strength, and machinability must be considered. The material that provides the most bending possible while still manufacturable should be selected. Taking the flexure shapes above, the material selected should be some form of EDM capable metal. The small whisker size means that other manufacturing methods could cause plastic deformation during the manufacturing process. While plastics could be molded to meet the flexure shape, plastics also creep which is

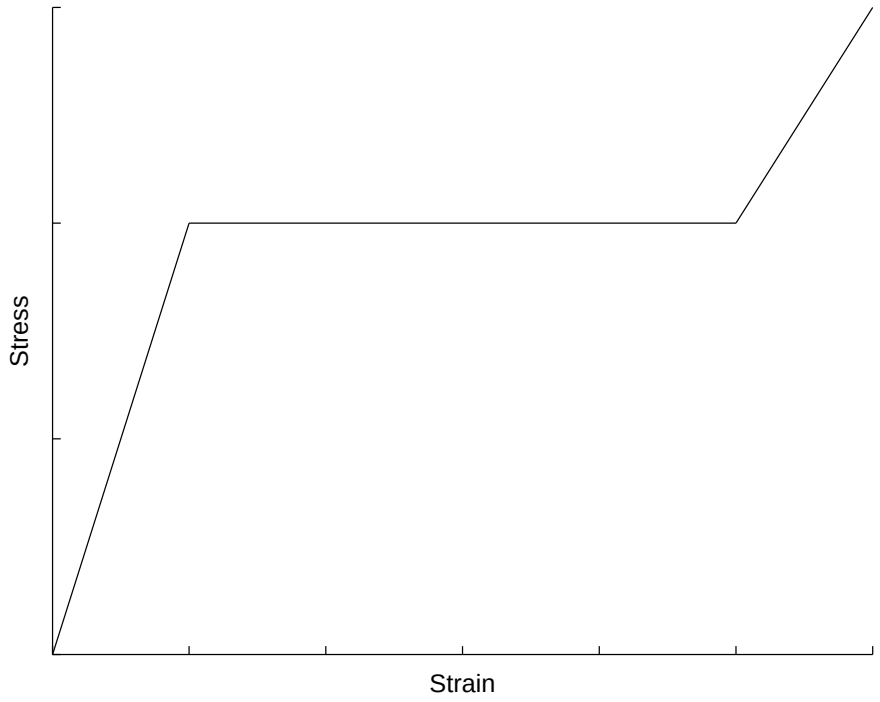


Figure 2.9: Stress-Strain Curve for a Superelastic Material

undesired. Creep is the deformation over time of the material due to an applied load. In addition, plastics have very poor crack propagation properties, making any crack or scratch on the surface a major problem for fatigue life. The modulus of elasticity and yield strength together set the total elastic deformation a material can undergo. For the flexure, the material that can deform the most without failure is desired, as that material will allow the VSA to rotate the most. To this effect, the flexure material selected was a superelastic nickel-titanium alloy, commonly referred to as nitinol.

Nitinol has a low modulus of elasticity with a relatively high yield strength, and is also a superelastic material. Figure 2.9 shows the stress-strain curve for a superelastic material. The high yield strength combined with the low modulus of elasticity allows it to undergo 0.9 percent deformation before reaching the plateau yield stress. In addition, the superelastic property allows it to undergo additional reversible deformation. A superelastic

material doesn't undergo plastic deformation when the plateau yield stress is reached, instead it undergoes a reversible change in phase for a large force range before plastically deforming. This superelasticity will allow nitinol to resist plastic deformation much more readily than conventional materials. In addition, the low modulus of elasticity allows the remainder of the VSA to be less stiff, decreasing the size of the remaining components.

2.3 Summary

In this chapter the design of the Arched Flexure VSA's flexure was presented. The flexure was defined by independent variables and then optimized for two different cases. It was optimized for maximum stiffness range, and also optimized for maximum stiffness range with constant relative sensitivity. The flexure optimized for maximum stiffness range resulted in a stiffness ratio of 1200, while the flexure optimized for maximum stiffness range with constant relative sensitivity resulted in a stiffness ratio of 100. The optimization was performed using a genetic optimization with proper constraints and bounds. The flexure optimized for maximum stiffness range with constant relative sensitivity was selected for manufacture and use in the VSA.

CHAPTER 3

DESIGN OF THE ARCHED FLEXURE VARIABLE STIFFNESS ACTUATOR

The Arched Flexure VSA is designed to provide a very large range of stiffness with constant relative sensitivity, and can provide a free joint. The flexure was optimized for two different cases: one for maximum stiffness ratio only and one for maximum stiffness ratio with constant relative sensitivity. The flexure optimization was performed with normalized variables, allowing the flexure to be applied to any relatively sized VSA.

Several key functions are associated with a VSA. Key components include the stiffness selection system, the joint actuation system, and the overall joint structure. The flexure design, described in Chapter 2, is a key sub-component of the stiffness selection system. The other sub-components of the stiffness selection system interface with the flexureto obtain the performance listed in Table 1.2. The stiffness selection system provides the connection between the flexure and the next link in the joint. They change the stiffness value by changing the contact point on the flexure, and transfer forces from the flexure to the next link. The joint actuation system moves the flexure and generates the torque needed to drive the connected link. The overall joint structure provides mounting for all components, including the adjacent links. It also constrains the VSA to a one degree of freedom rotational joint.

The overall design of the VSA is described first. Next, the overall stiffness of the design is compared to the flexure stiffness, to ensure that the component stiffness is significantly higher than the flexure stiffness. Then, the design of the stiffness selection system, joint actuation system, and the overall joint structure are described in detail. Finally, the desired specifications of Table 1.2 for the VSA are compared to those for the designed system.

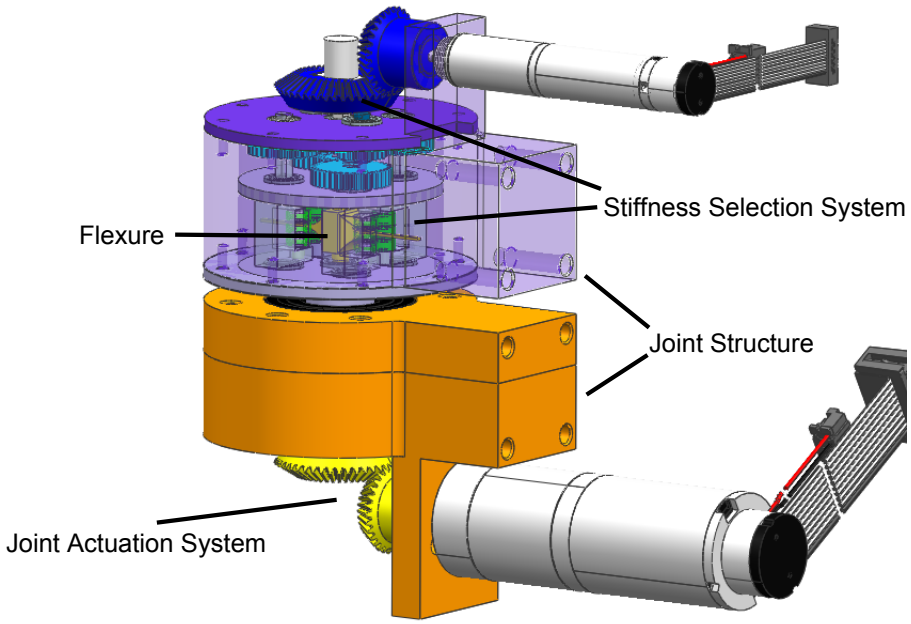


Figure 3.1: Overall CAD model of the VSA

3.1 Design of the Arched Flexure VSA System

The overall VSA design creates a variable compliance one degree of rotational freedom joint. It allows a wide range of operating stiffnesses about this one degree of freedom. Figure 3.1 shows the overall CAD model of the designed VSA. The overall VSA can be broken into several key systems which meet the VSAs designed goals. Those key systems are the stiffness selection system, the joint activation system, and the overall joint structure. The stiffness selection system provides the changing contact point with the flexure. It sets the stiffness that the VSA joint provides. The dimensionless flexure optimized in Chapter 2 is appropriately sized for a low-cost proof-of-concept prototype. The flexure provides the dimensions that the remainder of the systems must match. The stiffness selection system incorporates the flexure, and changes the stiffness of the VSA. The joint actuation system provides the power to the overall VSA. It changes the joint position and provides power to the joint. The overall joint structure provides mounting for all components and constrains

Table 3.1: Flexure Design Parameters for Manufactured Flexure

Variable	Value
X	0.2900 in
R_1	0.2715 in
R_2	0.0375 in
R_3	0.2484 in
h_f	0.0388 in
w_f	0.0199 in
θ_1	0.9608 rad
θ_2	0.7834 rad

the VSA to a one degree of freedom rotational joint. It also provides connections to the links.

The flexure optimized in Chapter 2 is sized for a proof-of-concept prototype. Since the optimization was performed using normalized variables, it can be applied to any relatively sized flexure, scaling off of R_2. This allows the flexure shape to be scaled to whichever application is required. For the proof-of-concept prototype the connector radius, R_2, is set to be 0.0375 inches. This was set to decrease the size of the VSA. See Table 3.1 for the variable values. The flexure chosen for manufacture and testing was the flexure optimized for relative sensitivity and stiffness ratio.

3.2 Overall Stiffness of the Design

The optimization performed in Chapter 2 describes a flexure with a stiffness curve given the stiffness selection system angle. To achieve overall VSA stiffness close to that, the elastic properties of the flexure must dominate the VSA stiffness for all force application angles. This means that the stiffness of all other components in the VSA must be significantly higher than the highest flexure stiffness. Since the flexure is mounted in series with the other components in the VSA, the equivalent VSA stiffness is given by:

$$k_{eq} = \frac{k_f * k_c}{k_f + k_c} \tag{3.1}$$

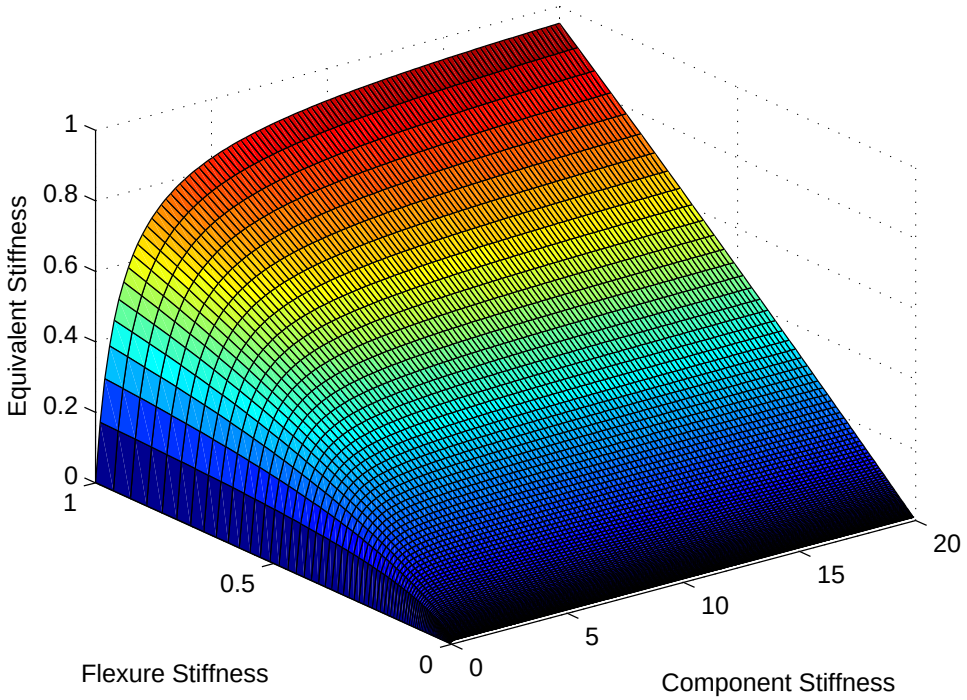


Figure 3.2: Overall VSA Stiffness vs Normalized Flexure Stiffness and Component Stiffness

where k_{eq} is the equivalent stiffness, k_f is the flexure stiffness, and k_c is the other component effective stiffness. The stiffness of the remaining components should be at least five times greater than the maximum stiffness of the flexure.

The minimum component stiffness of five times greater than the maximum flexure stiffness comes from applying Equation 3.1 through a set of component stiffness values and observing the resulting VSA stiffness. Applying a range of those values results in Figure 3.2 for the VSA stiffness. Figure 3.2 shows how the equivalent stiffness changes with force application angle and component stiffness. Since the flexure being manufactured was optimized for maximum stiffness while maintaining relative sensitivity, the component stiffness selected should be a function of relative sensitivity and overall stiffness as well. To assist with this, an exponential curve can be fit for each component stiffness and the least

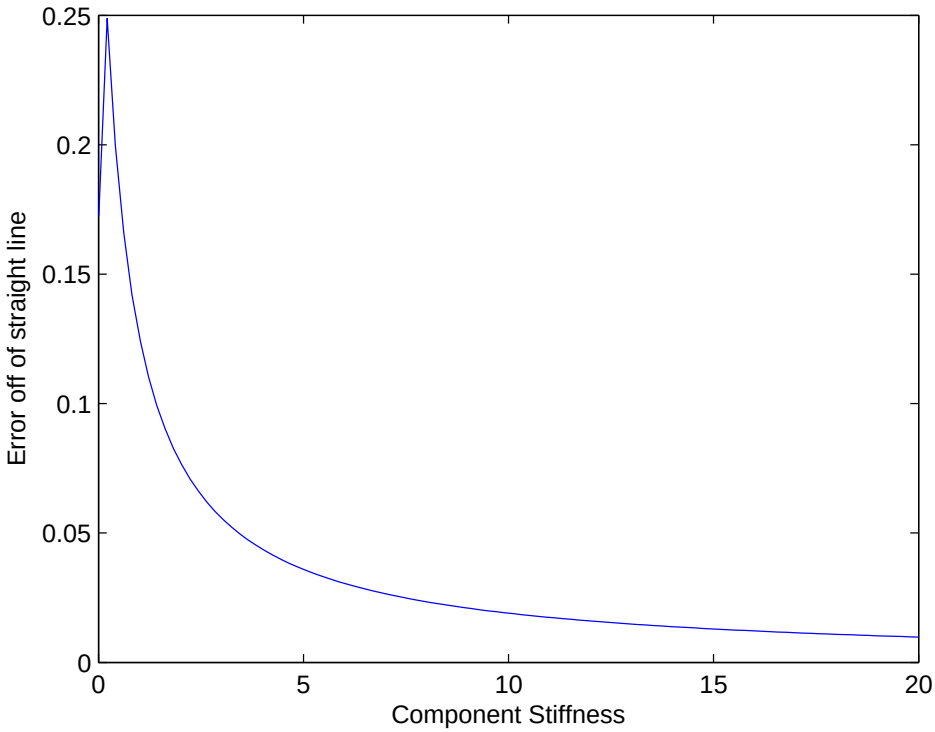


Figure 3.3: Least Squares Error vs Normalized Stiffness of Components

squares error can be calculated. This resulted in Figure 3.3, the least squares error values relative to the normalized stiffness of the components.

As the design is set, various components need to have their stiffness values calculated and compared to the minimum acceptable stiffness. While it is possible to use CAD programs to numerically calculate the overall stiffness of the device, the number of components and large number of rotation axis make this unfeasible. Therefore, the simple stiffness equations were used, which are written in Equations 3.2, 3.4, 3.3 for torsional, bending, and axial loading respectively.

$$k_r = \frac{G * J}{L} \tag{3.2}$$

$$k = \frac{48EI}{L^3} \tag{3.3}$$

$$k = \frac{EA}{L} \tag{3.4}$$

Equation 3.2 allows the rotational stiffness of a shaft to be determined, where G is the modulus of rigidity, L is the length of the shaft, and J is the area moment of inertia for the given shaft geometry. Equation 3.3 allows the stiffness of a beam to bending given a cantilevered support be determined, where E is Young's modulus, L is the length of the beam, and I is the area moment of inertia for the given beam geometry. Equation 3.4 defines the stiffness of a beam given axial loading, where E is Young's modulus, L is the length of the beam, and A is the area for the given beam geometry. Note that any linear stiffnesses must be converted to rotational stiffness based upon where the component is located, which occurs on a part by part basis.

3.3 Stiffness Selection System Design

The stiffness selection system provide the connection between the flexure and the next link, as well as providing the changing contact point on the flexure. The contact point changes the flexure stiffness and allows the VSA to rapidly change stiffness value. To create the stiffness selection system, several requirements must be maintained. First, the axis of rotation for the stiffness selection system is defined by the flexure optimization. The stiffness selection system axis of rotation must be at the center of curvature for the curved portions of the flexure. This restriction ensures that when the stiffness selection systems rotate they maintain contact with the flexure at all times. Second, the total length from the axis of rotation to the outside end point of the stiffness selection system must be equal to R_1. This restriction ensures that the stiffness selection system make contact with the flexure without deforming the flexure. Third, the stiffness selection system must maintain a line of contact that changes the effective length of the beam. It ranges from w_f to a length of 0.3 inches. This restriction ensures that the entirety of the flexure contacts the stiffness selection system, keeping the assumptions made in the FEA valid. See Figure 3.4 for a schematic

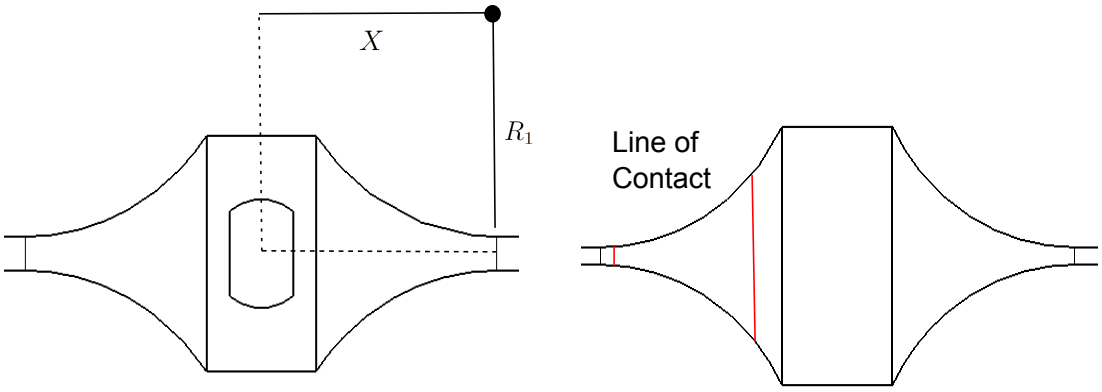


Figure 3.4: Schematic Representation of Stiffness Selection System Requirements

representation of these requirements. The stiffness selection systems must rotate in two different directions to maintain the proper contact with the flexure. Figure 3.5 shows the directions of rotation for the four stiffness selection systems.

The stiffness selection system was also designed so it has a low friction connection between the flexure and the remainder of the system. Since the flexure and the stiffness selection system transmit the torque for the joint, the connection point between them may have to handle a large amount of force. Without a low friction connection, the small motor could have difficulty overcoming friction to change the stiffness. The low friction connection must also have a radius smaller than R_1 at the point of contact.

The stiffness selection systems consist of a central shaft with two bar shaped offshoots. The offshoots connect to a shaft with bearings. The bearings physically contact the flexure and provide a low friction connection. See Figure 3.6 for a CAD model of the stiffness selection system. Figure 3.7 shows a CAD model of the stiffness selection system gearing for the small motor. The stiffness selection system shown in Figures 3.6 and 3.7 was designed and meets the requirements. The low friction contact was a steel sleeve that ensures the flexure is always contacted at any point, and that no gaps in contact occur. The sleeve allows rotation about the center shaft, meaning that the motor moving the stiffness selection system does not need to overcome large forces due to friction. The sleeve also

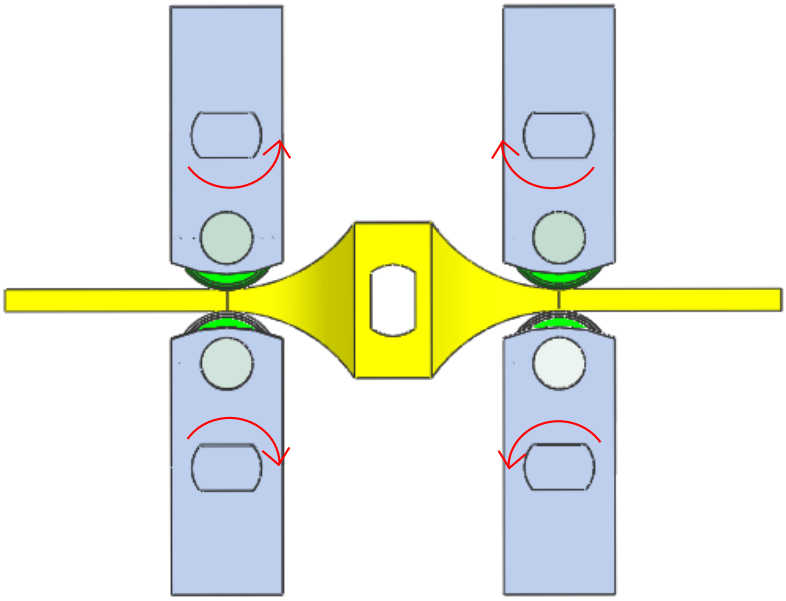


Figure 3.5: Schematic Representation of Rotation Direction for Stiffness Selection Systems

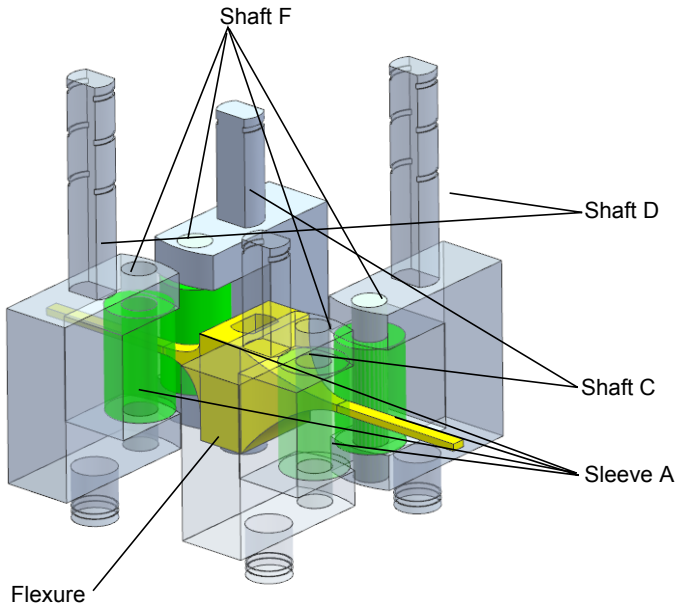


Figure 3.6: CAD model of Stiffness Selection System without Gearing

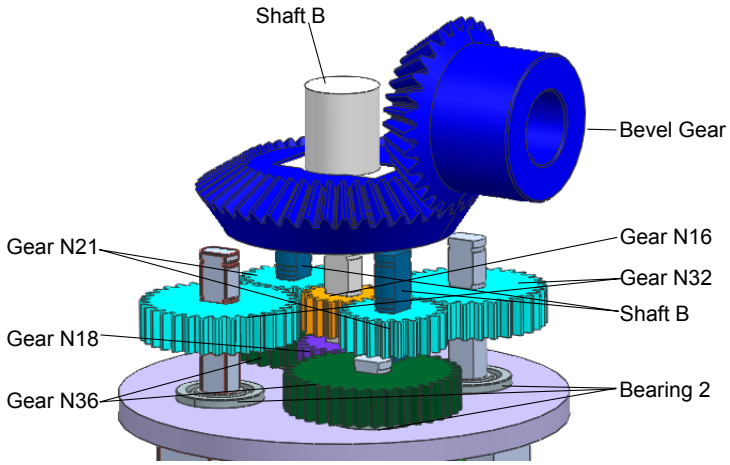


Figure 3.7: CAD model of Stiffness Selection System Gearing

helps with the assembly process. Finally, it helps ensure that the stiffness of that component is high, using bearings would make the perceived stiffness of the VSA much lower. The bar shaped offshoots had a curved radius of R_1 on the end. This curved radius ensures that the bars do not intersect each other or the flexure as they rotate.

The gear ratio between the central shaft for the motor and the stiffness selection system is 1:2, which allows for a wide range of possible options for gears. A 36 tooth gear is used for the contactor side, and a 18 tooth gear is used for the central shaft, both have a diametral pitch of 64. These gears provide the gear ratio of 1:2 and also have the correct pitch diameters to ensure that the stiffness selection system axis of rotation would not move. The other two contactors use a 32 tooth gear with a 21 tooth idler gear and 16 tooth gear on the central shaft. Note that the idler gear location not only needs to be in a mountable location, but the idler gear bearings and gear shaft bearings are able to fit in those locations as well. Figure 3.8 shows that the shafts and bearings do not intersect each other. Note that the two different colors represent the location of the holes on separate planes.

The bevel gears have a ratio of 1:1.5. Flats on the bevel gears provide the torque transfer to the shafts. In addition, one bevel gear had the hub cut off to decrease the space that it occupied, making the VSA more compact. Connectors were designed that took the motor output shafts and connected them to the bevel gears. All of the selected gears and

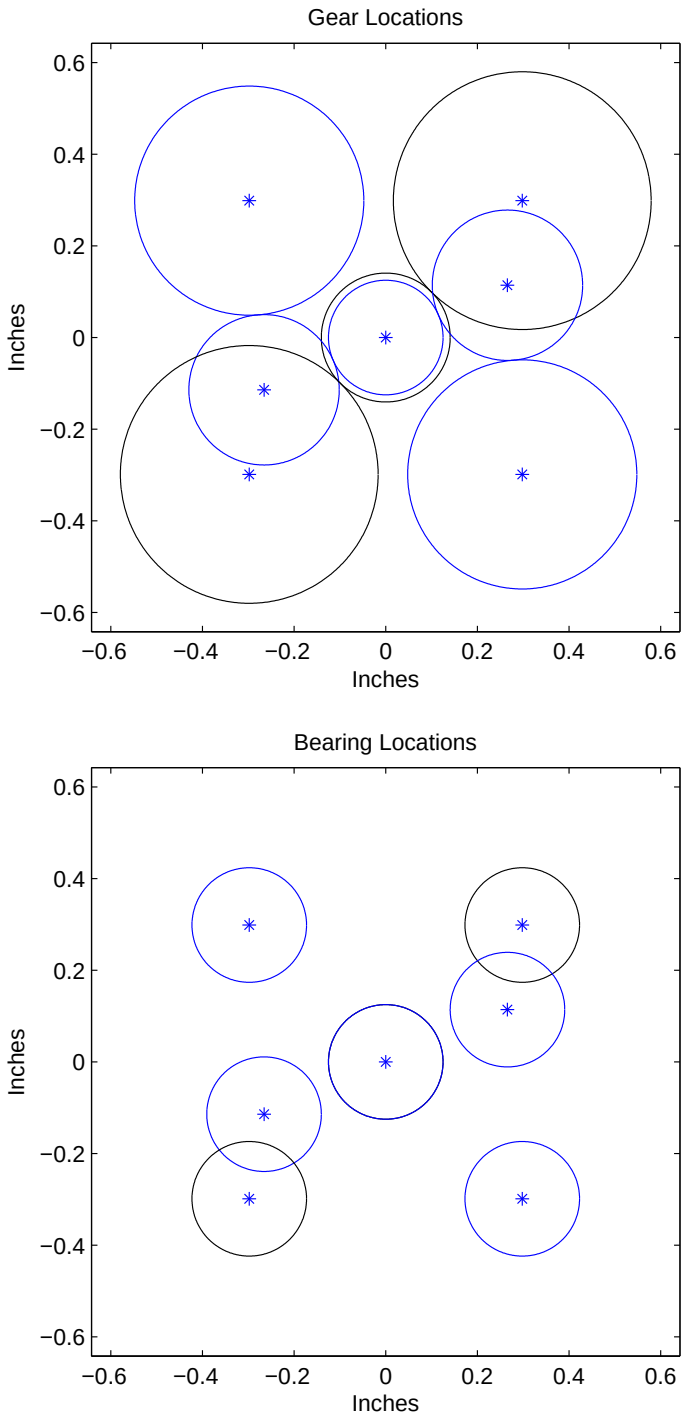


Figure 3.8: Stiffness Selection System Shaft, Gear, and Bearing Placement

Table 3.2: Stiffness Selection System Parts List

Part	**Quantity**
Bearing 2	14
Bevel Gear	2
Flexure	1
Gear N36	2
Gear N32	2
Gear N21	2
Gear N18	1
Gear N16	1
Maxon DCX10L Motor	1
Sleeve A	4
Shaft B	1
Shaft C	2
Shaft D	2
Shaft E	2
Shaft F	4
Small Motor Connector	1

shafts used double D connections, which means that the shafts had flats cut into them and the gears had that shape cut instead of a round hole for the shaft. This allowed the shafts to be connected to the gears without keys or splines.

The small motor (a Maxon DCX10L) is used to rotate the stiffness selection system. This motor provides 3.06 Watts of power, and its integrated gear head provides a gear ratio of 64:1. The total gear ratio of 192:1 between the motor and the stiffness selection system allows the low power motor to move the stiffness selection system. The motor's maximum speed of 11900 RPM moves the stiffness selection system from minimum stiffness to maximum stiffness in 0.121 seconds. This does not meet the ideal specification but instead meets the marginal specification, while still keeping the cost down. See Appendix B for the drawings of the custom parts mentioned above. Table 3.2 provides a list of parts making up the stiffness selection system.

Shaft F, the shaft connecting the bearings to the stiffness selection system, shown in Figure 3.6, is analyzed with Equation 3.3, due to the simply supported bending. Taking the

relevant variables from the drawing, the stiffness results in:

$$k_F = \frac{L^3}{48EI} = 6.77 * 10^7 \text{N/m} \tag{3.5}$$

Equation 3.5 must then be converted to a representative angular stiffness. To do this, the stiffness is multiplied by the angular change that would occur at the maximum stiffness location, which results in Equation 3.6.

$$k_R = 999.8 \text{Nm/rad} \tag{3.6}$$

Equation 3.6 can then be compared to the flexure's stiffness, and the ratio realized is 10. Since this ratio is greater than five, shaft F meets the design's minimum stiffness requirements.

The offset bar on the stiffness selection system must be analyzed with Equation 3.2, due to the bending with cantilevered support. Taking the relevant variables from the drawing, the stiffness results in Equation 3.7.

$$k_{bar} = \frac{EA}{L} = 5.16 * 10^7 \text{N/m} \tag{3.7}$$

Equation 3.7 must then be converted to a representative angular stiffness. To do this, the stiffness is multiplied by the angular change that would occur at the maximum stiffness location, which results in Equation 3.8

$$k_R = 761.5 \text{Nm/rad} \tag{3.8}$$

Equation 3.8 can then be compared to the flexure's stiffness, and the ratio realized is 7.6. Since this ratio is greater than five, offset bar on the stiffness selection system meets the design's minimum stiffness requirements.

Table 3.3: Joint Actuation System Parts List

Part	**Quantity**
Bearing 3	2
Bevel Gear	2
Connector Motor Large	1
Maxon DCX22S Motor	1
Shaft A	1

3.4 Joint Actuation System Design

The joint actuation system drives the flexure. To make the VSA more compact, the motor is mounted parallel to the links. The joint actuation system consists of the large motor, a bevel gear, bearings, and a shaft to connect to the flexure. A shaft of initial diameter of 0.125 inches, with the double D shape, is used to match the shape of the flexure mounting location, but is thicker along its length (diameter of 0.25 inches) to meet stiffness requirements. The large motor (a Maxon DCX22S) is used to move the joint. This motor provides 23.2 Watts of power, and it's integrated gear head provides a total gear ratio of 364.5:1 between the motor and the flexure. The motor's maximum speed of 12400 RPM means that the VSA can move at a speed of 34RPM. See Appendix B for the drawings of the custom parts mentioned above as well as the modified parts. See Table 3.3 for the list of parts making up the large joint actuation subsystem.

Shaft A, the shaft connecting the flexure and the bevel gear attached to the large motor, must be analyzed with Equation 3.2, because of the torsional load. Taking the relevant variables from the drawing, the stiffness results in Equation 3.9.

$$k_A = \frac{G * J}{L} = 628\text{Nm/rad} \tag{3.9}$$

Equation 3.9 can then be compared to the flexure's stiffness, and the ratio realized is 6. Since this ratio is greater than five, shaft A meets the design's minimum stiffness requirements.

3.5 Joint Structure Design

The overall joint structure constrains all of the other systems, ensuring they are both protected from foreign objects as well as maintaining their proper locations. It also ensures the design operates as intended and works as a one degree of freedom revolute joint. The overall joint structure provides connections for the external links on the robot.

The overall joint structure ensures that the axis of rotation for the VSA as a whole is along the axis of rotation for the flexure. This restriction ensures that when the flexure bends the VSA rotates with the flexure only about the joint axis. The joint structure provides mounting locations for the connected links. The joint structure is significantly more stiff than the flexure's maximum stiffness, allowing the compliance of the flexure to dominate. The joint structure must also provide access to the flexure for proper assembly.

The overall joint structure consists of several plates, three hollow cylinders with offset plates for motor mounting, and bearings. The restriction to a one degree of freedom joint is created by constraining a bearing between two of the cylinders. The cylinders clamp the bearing preventing motion in all directions except rotation along the bearing's axis. The two cylinders are on the drive motor side of the VSA. This side has extra space to match the small motor side of the VSA. The third cylinder, the cylinder on the small motor side of the VSA, contains the flexure, stiffness selection system, and gearing for the small joint actuation system. It also contains plates that hold bearings for those systems. The plate closest to the large motor side of the VSA has a hollow shaft perpendicular to it, which fits into the clamped bearing. Figure 3.9 shows a CAD model of the overall joint structure.

In addition, the overall joint housing includes fixturing locations to ensure stiffness selection system orientation. Figure 3.10 shows the fixuring location. The fixuring locations consist of a simple slots machined into the housing cylinder. A metal bar placed between opposing slots is used to constrain the connector bars when the joint actuation system is assembled.

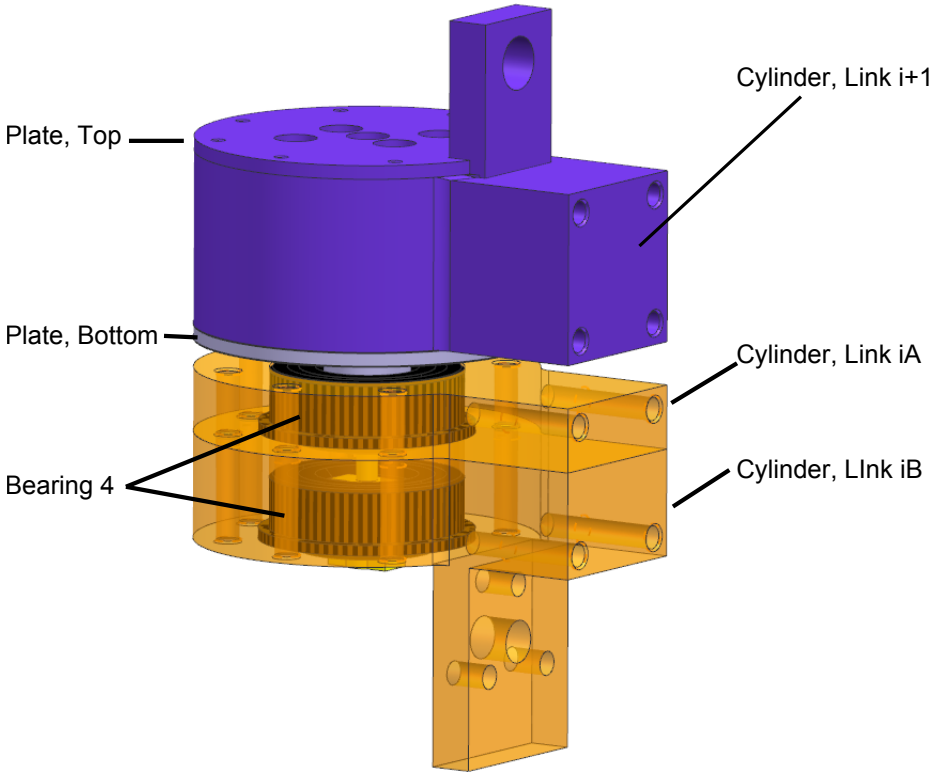


Figure 3.9: CAD model of Overall Joint Structure

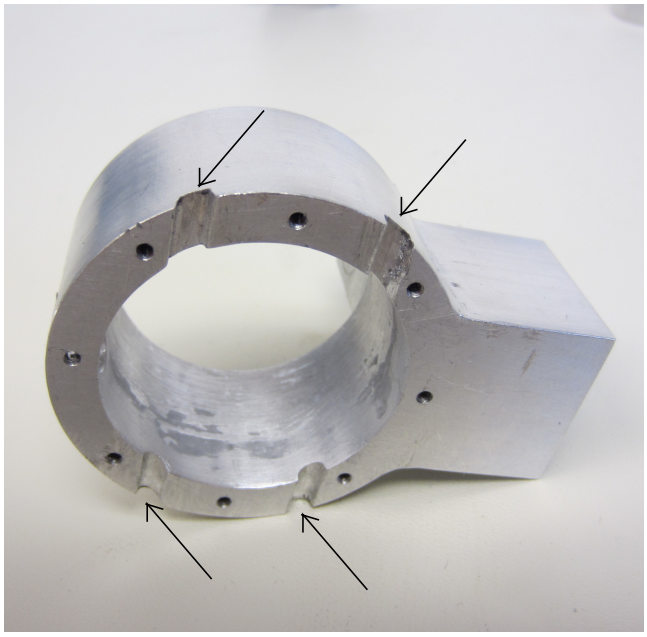


Figure 3.10: Fixuring Locations

Table 3.4: Overall Joint Structure Parts List

Part	**Quantity**
Bearing 4	2
Cylinder, Link i+1	1
Cylinder, Link i A	1
Cylinder, Link i B	1
Plate, Bottom	1
Plate, Middle	1
Plate, Top	1

The overall joint structure shown in Figure 3.9 also shows the link connection method. The link connection method consists of a flat extended section with 1 inch by 1 inch dimensions and four 8-32 screw holes to mount into. The section extends sufficiently past the remainder of the VSA to ensure links can connect to it. See Appendix B for the drawings and geometric tolerances of the custom parts mentioned above. See Table 3.4 for the list of parts making up the overall joint structure.

3.6 Results

The Arched Flexure VSA meets the requirements set by the flexure optimization and the additional requirements for overall VSA functionality. Table 3.5 shows the full bill of materials for the VSA design. The off-the-shelf parts can be purchased and used directly in the design. Manufactured parts must be manufactured from metal stock of various types. The type of metal is set in the drawing on a part by part basis, for example the shafts were all machined out of steel.

3.6.1 Physical Assembly

Figure 3.11 is a picture of the Arched Flexure VSA fully assembled. Note that the working components cannot be seen as they are housed within the overall joint structure. Figure 3.12 is a picture of the internal components of the VSA. The VSA was assembled using the

Table 3.5: Bill of Materials

Part	**Quantity**	**Supplier**	**Part Number**
Bearing 2	14	Grainger	1ZEZ6
Bearing 3	2	Grainger	1ZEX2
Bearing 4	2	McMaster-Carr	6383k234
Bevel Gear	4	SDP/SI	S1346Z-48S30S045
Connector Motor Small	1	Manufactured	
Connector Motor Large	1	Manufactured	
Cylinder, Link i+1	1	Manufactured	
Cylinder, Link i A	1	Manufactured	
Cylinder, Link i B	1	Manufactured	
Flexure	1	Manufactured	
Gear N36	2	Manufactured	
Gear N32	2	Manufactured	
Gear N21	2	Manufactured	
Gear N18	1	Manufactured	
Gear N16	1	Manufactured	
Maxon DCX10L Motor	1	Maxon Motors	B717A4FF3FB0
Maxon DCX22S Motor	1	Maxon Motors	B717A4FF3D3E
Plate, Bottom	1	Manufactured	
Plate, Middle	1	Manufactured	
Plate, Top	1	Manufactured	
Screw A	7	McMaster-Carr	91255A199
Screw B	15	McMaster-Carr	90910A114
Screw C	6	Menards	87235
Shaft A	1	Manufactured	
Shaft B	1	Manufactured	
Shaft C	2	Manufactured	
Shaft D	2	Manufactured	
Shaft E	2	Manufactured	
Shaft F	4	Manufactured	
Sleeve A	4	Manufactured	
Snap Ring A	23	McMaster-Carr	97633A110
Snap Ring B	3	McMaster-Carr	97633A130
Snap Ring C	1	McMaster-Carr	97633A200

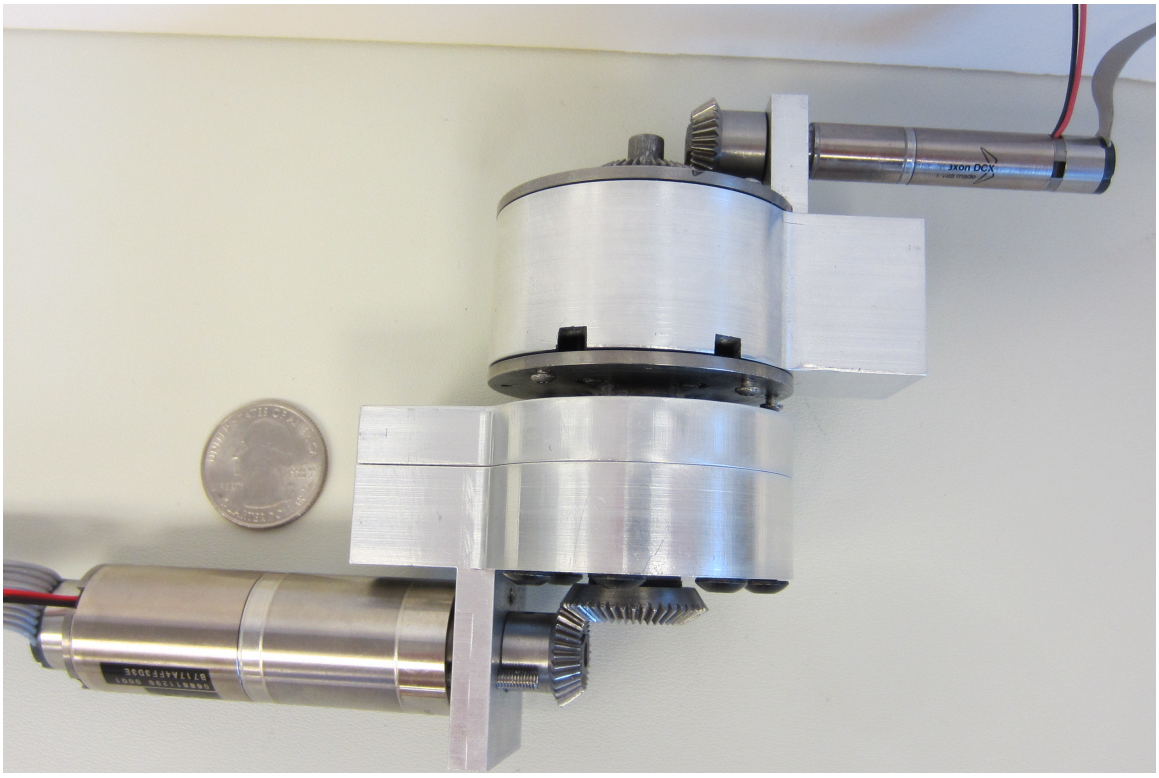


Figure 3.11: Overall Manufactured VSA

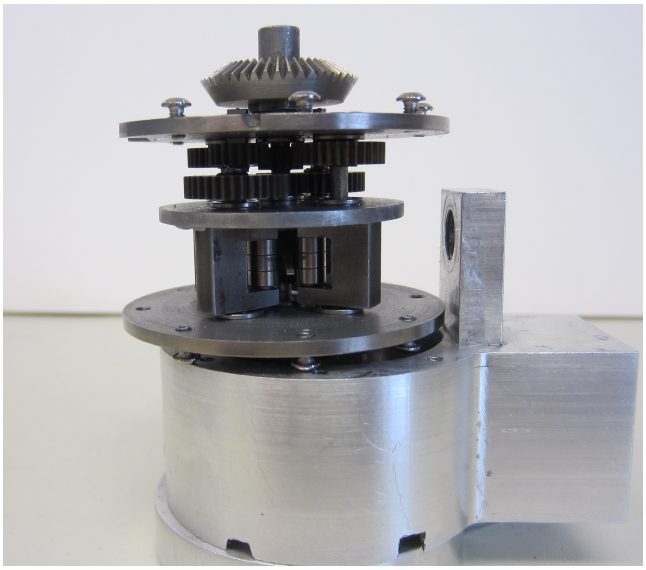


Figure 3.12: Internal Components of the VSA

assembly plan provided in Appendix B.

3.7 Summary

The Arched Flexure VSA consists of the stiffness selection system, joint actuation system, and overall joint structure. The flexure optimized in Chapter 2 was appropriately scaled for the intended application. The stiffness selection system allows the joint stiffness to be controlled. The joint actuation system provides the power to the overall VSA. It changes the joint position and provides power to the joint. The overall joint structure provides mounting for all components and constrains the VSA to a one degree of freedom rotational joint. It also provides connections to the links. The physical prototype will be tested to confirm that the experimental results match the theoretical results, showing that the VSA functions as intended.

CHAPTER 4

EXPERIMENTAL VERIFICATION

The overall VSA design, described in Chapter 3, represents a VSA optimized for maximum stiffness range with constant relative sensitivity. It can also provide a free joint. The flexure, described in Chapter 2, provides the elastic behavior for the VSA. The VSA has several key systems, specifically the stiffness selection system, the joint actuation system, and the overall joint structure. Each of these systems help the VSA meet the design specifications that describe a good VSA, provided in Chapter 1.

The Arched Flexure VSA design performance is experimentally verified to ensure that it meets design specifications.

First, the VSA design specifications developed in Chapter 1 are reviewed. Next, VSA performance is reviewed for design aspects that do not require rigorous testing. The design performance of each of the design aspects that require testing are then reviewed. Both test procedures and design performance are described.

4.1 Specifications Review

The overall VSA design, described in Chapter 3, has a theoretical stiffness ratio of 100 and constant relative sensitivity. This design's performance must be experimentally verified to ensure that it meets the expected values. The design specifications from Chapter 1, seen in table 1.2, must be confirmed.

Experiments were designed to test the design specifications. The experiments were designed to limit the forces imposed on the flexure at low stiffness values to prevent plastically deforming the flexure. The same concept applies for force loads on bearings during testing, they cannot exceed the bearings rated forces.

4.2 Initial Verification

Several of the design criteria can be easily tested and confirmed without significant experimental setup. The Arched Flexure VSA weighs 1.44 pounds, which is very close to the expected weight of 1.43 pounds and exceeds the ideal value for weight. This low weight will allow the VSA to be applied to many different applications and decreases the inertia that robots need to overcome when using the VSA.

The Arched Flexure VSA exceeds the marginal size value with a size of 4.5 inches by 2 inches by 5 inches. This is larger than the ideal size value, but it does meet the marginal value and keeps the VSA design compact. The compact size allows the VSA to be applied to various applications that have size constraints.

The Arched Flexure VSA changes stiffness in 0.12 seconds, exceeding the marginal value for variation time. This does not satisfy the ideal value, but it does meet the marginal value, and is very close to the ideal value. The VSA could have a smaller gear ratio to decrease this time, but that was limited by the prototype cost requirement.

The Arched Flexure VSA provides 360 degrees of motion, meeting the ideal value for range of motion. This provides a major benefit to the design, allowing the design to directly replace conventional actuators.

4.3 Experiments Performed

Two different experiments were run to test all of the remaining criteria. They each require different experimental setups and data recording systems. The maximum stiffness, minimum stiffness, and stiffness at all angles requirements were evaluated in a single experimental setup. The free joint range at different contactor angles was tested in a separate experiment.

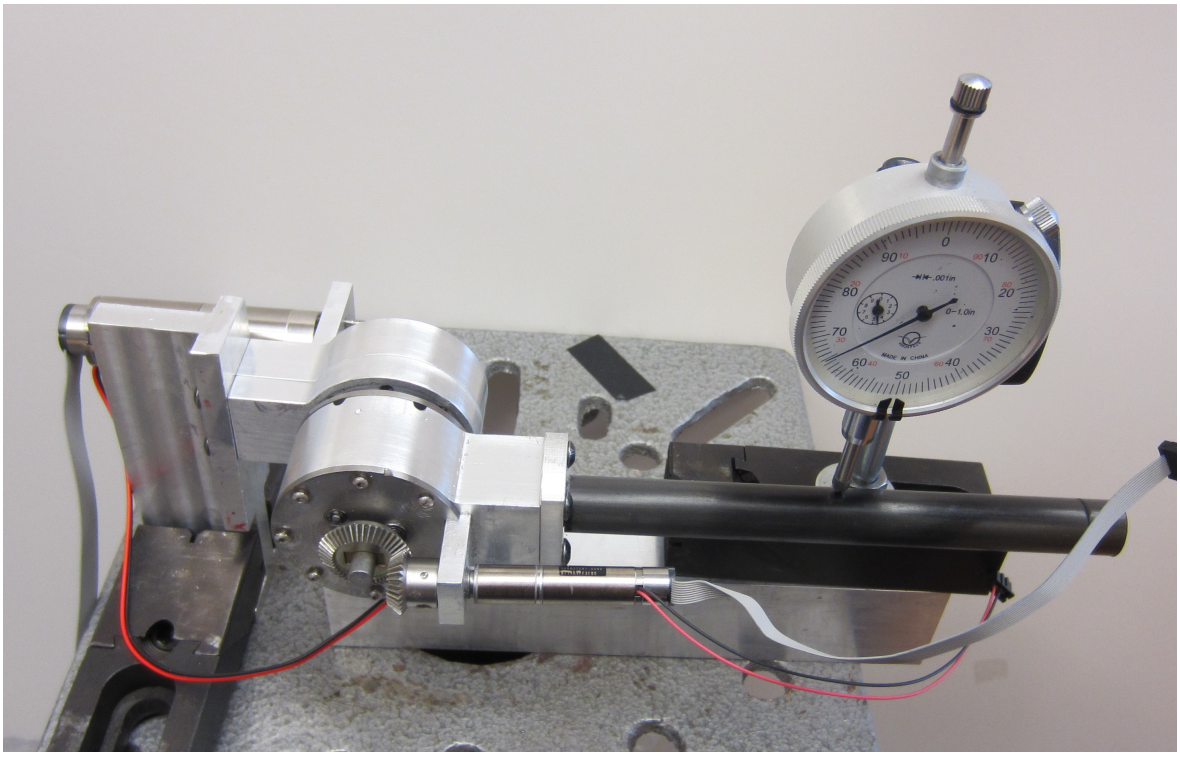


Figure 4.1: Picture of Experimental Setup Measuring Joint Stiffness

4.3.1 Stiffness Verification Experiment

The stiffness verification experiment was designed to confirm the maximum stiffness, minimum stiffness, and stiffness at intermediate angles. To gather this information, the following experiment was performed. One side of the VSA was rigidly mounted to a fixed location with the axis of rotation **perpendicular** to gravity. This allows gravity to provide the force used in calibration. A carbon fiber tube was attached to the other side of the VSA. This provides the moment arm that converts gravity supplied to the VSA to torque. A dial indicator was used to measure the distance that the link descends due to the applied load. Figure 4.1 shows the experimental setup. In this case, the moment arm has a total length of 8 inches to the torque source and the dial indicator records link deflection at 5 inches from the joint axis of rotation.

The dial indicator is mounted at 5 inches to provide an amplification of the angular displacements that occur when torques are applied to the VSA. This displacement

information can be converted to angular displacements using

$$\psi = \arcsin(\frac{\Delta y}{5}) \tag{4.1}$$

where Δy is the deflection at the dial indicator in inches, and ψ is the angular displacement in radians. The joint stiffness is calculated using

$$k = \frac{F * 8 * 25.4}{1000 * \psi} \text{Nm/rad} \tag{4.2}$$

where k is the stiffness, F is the force applied to the end of the link, and ψ is the angular displacement from Equation 4.1.

Equation 4.2 calculates the stiffness associated with the measured displacement and torque applied. Multiple measurements were taken for each contact selection angle. A set of 10 masses (7 grams each) and 10 link deflections were obtained for each contactor angle. To determine the stiffness, the least squares fit curve was generated for torque vs angular position of the stiffness selection system. The slope of the line is the average stiffness value for that contact selection angle. Figure 4.2 shows this for a sample dataset.

The above procedure was repeated for the set of contact selection angles ranging from 0 to θ_2. Figure 4.3 illustrates the variation with contactor angle. The stiffness results show that it follows the expected exponential pattern. The VSA provides a stiffness ratio of 55, which is less than the expected value of 100. Figure 4.3 also illustrates the theroretical behavior of the flexure alone. Comparison of the curves shows that the actual stiffness follows the same type of curve as the theoretical results, with a steadily increasing deviation as contactor angle increases.

4.3.2 Free Joint Range Verification Experiment

The second experiment was designed to confirm the free joint range. To gather this information, one side of the VSA was rigidly mounted to a fixed location with the axis of

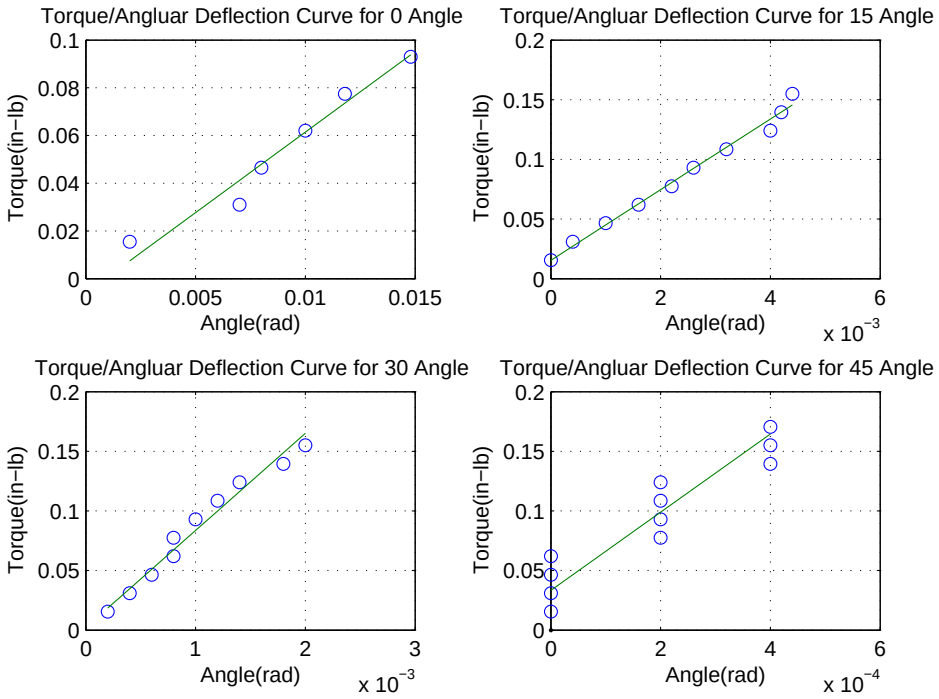


Figure 4.2: Torque vs Angular Position

rotation **parallel** to gravity. A carbon fiber tube attached to the VSA was used to transform angle changes to observable translational deflection. A piece of paper placed below the link was used to mark link deflection. Figure 4.4 shows the experimental setup.

The VSA contactor angle was set. The link was then rotated and the free rotation endpoints were marked on the sheet. The distance between the two endpoints is the chord length, c, an indicator of the free angle. Equation 4.3 shows the conversion from chord length to angle.

$$\theta = 2\arcsin(\frac{c}{2r}) = 2\arcsin(\frac{c}{2*8})\text{rad} \tag{4.3}$$

where c is the chord length in inches, and r is the distance to the joint axis of rotation in inches.

Figure 4.5 is a plot of the free joint range vs contactor angle compared to the predicted free joint range. Note that the experimental data closely follows the theoretical curve.

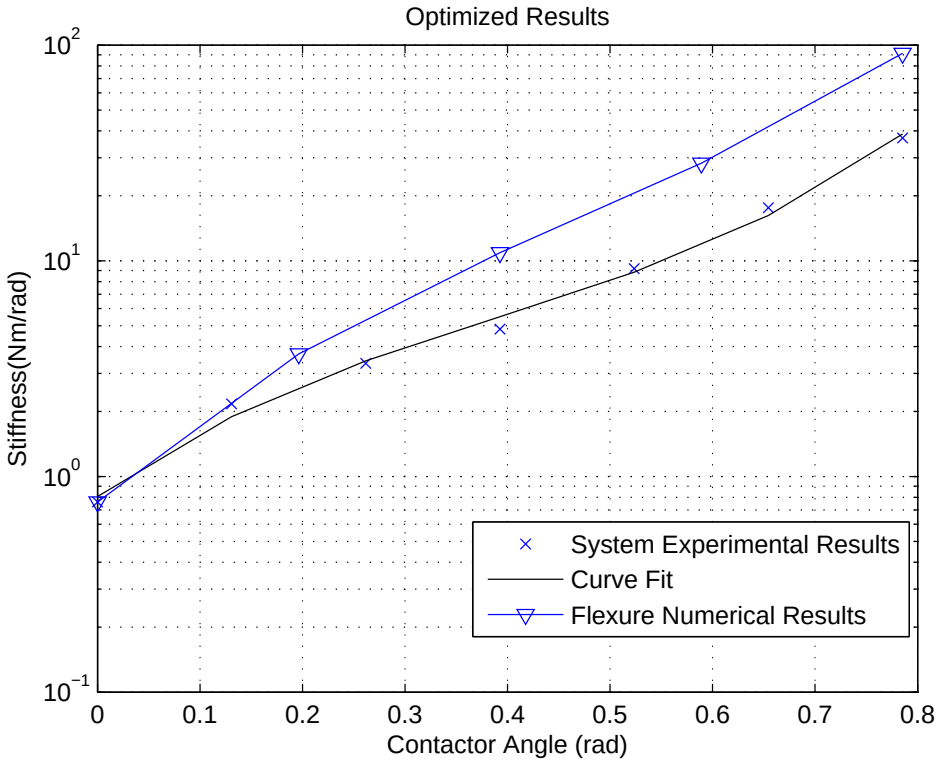


Figure 4.3: Angular Position vs Stiffness for Experimental Data

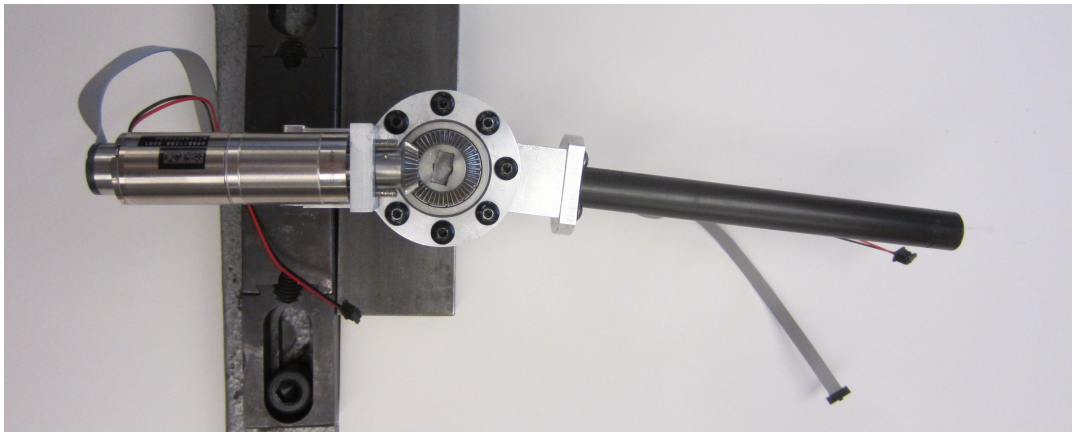


Figure 4.4: Picture of Experimental Setup Measuring Free Joint Range

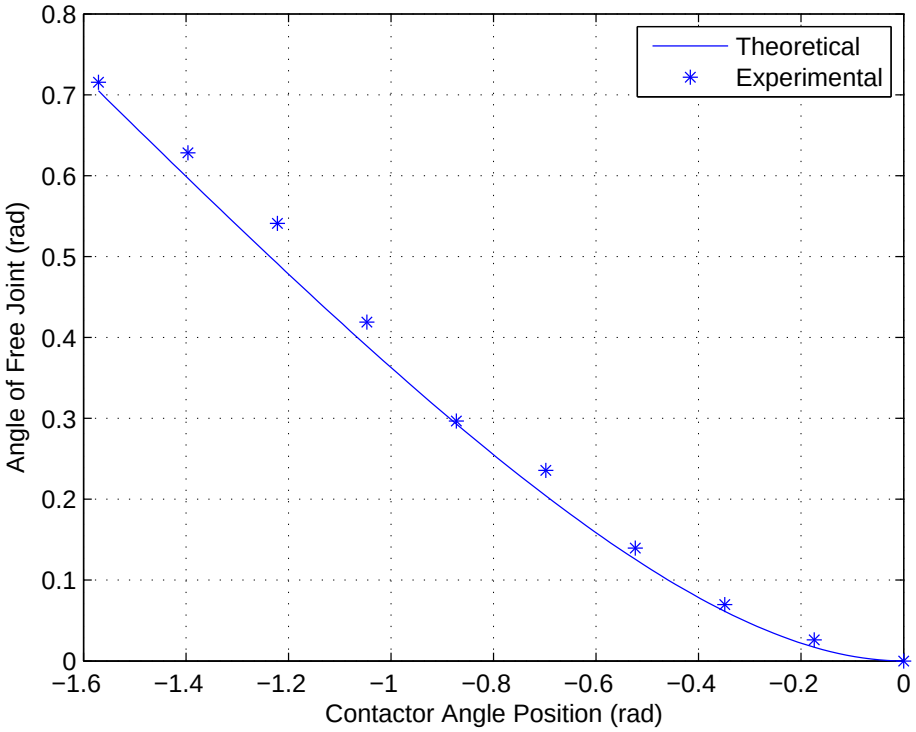


Figure 4.5: Contactor Angular Position vs Free Joint Range for Free Joint Angles

4.4 Discussion of Results

The discrepancy between beam theoretical stiffness and actual VSA stiffness is likely due to the compliance of the rest of the VSA system. The lowered stiffness value indicates that additional analysis should be performed. The experimental data can be compared to the theoretical flexure stiffness curve. Equation 4.4, the series spring equation, can be used to perform this comparison.

$$k_c = \frac{k_f * k_{eq}}{k_f - k_{eq}} \tag{4.4}$$

where k_f is the theoretical stiffness, k_{eq} is the experimental stiffness, and k_c is the component stiffness (combined stiffness of all VSA components that exist in series with the flexure). A component stiffness for each experimentally collected data point can be calculated using the interpolated theoretical stiffnesses and Equation 4.4. From these component stiffnesses, the constant component stiffness was selected by using the average component stiffness for the design. The constant component stiffness of 26.6 Nm/rad is established and matches the high and low theoretical values.

The difference between the theoretical and experimental stiffness is due to the reduced constant component stiffness. It also indicates the VSA component stiffness is significantly lower than the anticipated component stiffness of 500 Nm/rad. Taking the component stiffness and applying it to each data point yields a calibrated flexure stiffness curve that matches the theoretical flexure stiffness curve for the high and low values. The calibrated flexure stiffness curve matches the middle flexure values less, but still brings them closer to the theoretical stiffness. Figure 4.6 shows the calibrated flexure stiffness superimposed over the experimental system stiffness and theoretical flexure stiffness data. Note that the component stiffness has a greater impact on the larger flexure stiffness values, as it dominates the stiffness of the system (as it is lower than the flexure stiffness at that point). There are two possible reasons the middle experimental flexure points don't match the theoretical points. It could be due to an error in the theoretical curve at that point, or due

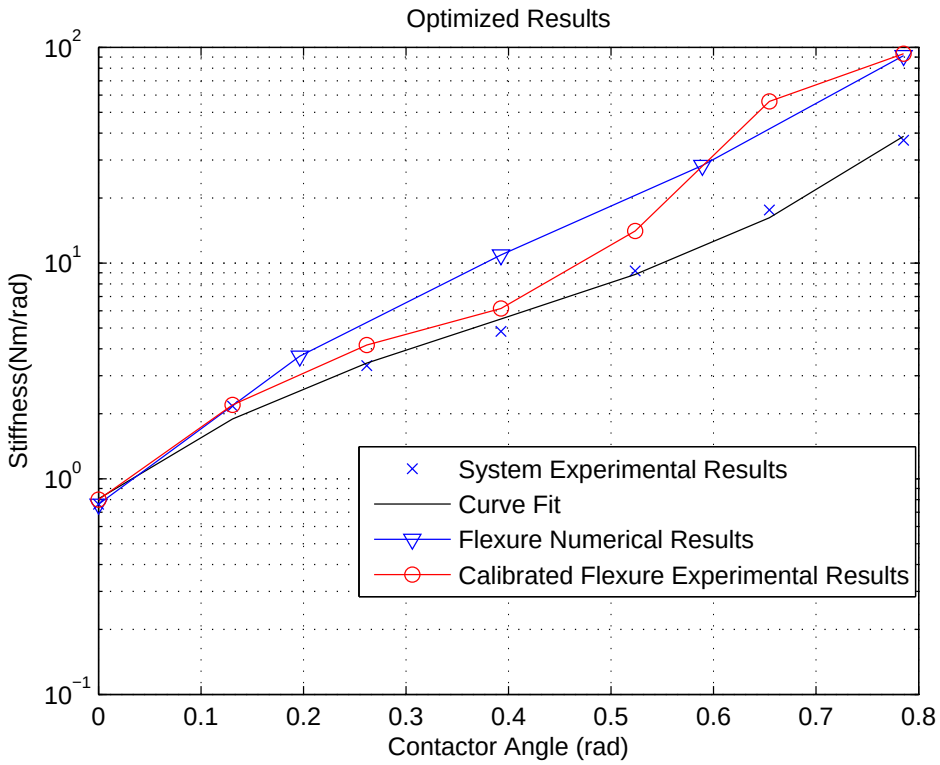


Figure 4.6: Stiffness-Angular Position Results

to a manufacturing defect in the shape of the flexure curve at that location. The manufacturing defect seems more likely, as the curves match for the remainder of the VSA points.

4.5 Proposed Design Revisions

The reduced VSA system stiffness could be caused by several different things. First, there were several cantilevered components in the VSA that could cause additional compliance in the system. Two of the connector bars in the stiffness selection system, the flexure, and two idler gears were all cantilevered. Each of these components could have been mounted against a plate with a bearing, but were not to decrease the size of the VSA. There could be bending in addition to rotational strains. For future designs, the cantilevered components should be constrained on both sides. The improved part constraints will increase the size of

the VSA. Second, the bearings chosen for the VSA were not explicitly analyzed for stiffness. Future designs should use stiffer and higher precision bearings that connect the stiffness selection systems to the overall joint structure. Third, the shaft connecting the flexure could have a larger diameter, and could connect to the flexure both through a central contact and by cupping the sides of the flexure. This would increase the stiffness of the shaft holding the flexure. Finally, making the flexure out of Nitinol instead of titanium would increase the relative stiffness of the system. Nitinol is around three times less stiff than the titanium. This change would make the relative stiffness of the system around three times greater.

Finally, the data shows that optimizing for relative sensitivity should not be included in the defining criteria for the joint. Since the VSA will have the theoretical stiffness reduced by the system stiffness, either the system stiffness needs to be set for the optimization, or the relative sensitivity component can be ignored. While system stiffness could be set, physically achieving the anticipated stiffness becomes impractical. Therefore the relative sensitivity component should be ignored for the VSA, and the maximum stiffness ratio optimized flexure should be used. In addition to the relative sensitivity term not having a major impact on the VSA as a whole, the flexure optimized for maximum stiffness ratio has a smaller minimum stiffness value. This would increase the stiffness ratio seen by the VSA, even if all component stiffnesses are maintained at the same level. Therefore, any future designs should use the flexure optimized for maximum stiffness ratio only.

4.6 Summary

The performance of the Arched Flexure VSA with respect to all of the design specifications is summarized in Table 4.1. All design specifications were satisfied except for stiffness ratio. The VSA had a stiffness ratio of 55, which is significantly lower than the anticipated ratio of 100.

Table 4.1: Experimental Results

Metric	**Ideal**	**Marginal**	**Result**
Stiffness Ratio	≥ 1000	≥ 100	55
Range of Motion	∞	180°	360 Degrees
Full Range Variation Time	0.1s	0.2s	0.12s
Size	$\leq$ 3in x 3in x 3in	$\leq$ 5in x 5in x 5in	4.5in x 2in x 5in
Prototype Cost	$\leq$ $ 5000	$\leq$ $ 10000	$ 4000
Weight	$\leq$ 2lb	$\leq$ 5lb	1.44 lb
Minimum Stiffness without a Free Joint	No Spec	No Spec	0.67Nm/rad
Maximum Stiffness	No Spec	No Spec	37Nm/rad

CHAPTER 5

SUMMARY AND FUTURE WORK

The performance of the Arched Flexure VSA is compared with other existing approaches. Recommendations for design improvements are also provided together with suggestions to future work in this area.

5.1 Performance Comparison with Existing Designs

Current conventional robots require high stiffness joints to provide absolute positioning accuracy in free space. The high stiffness joints limit what a robot can do in constrained manipulation. Robots can circumvent these problems though a variety of methods. Use of Variable Stiffness Actuators (VSAs) appear to be promising. VSAs provide commanded variable stiffness at a joint to suit the task being performed. Current VSA performance is summarized in Table 5.1, including the experimental results from the Arched Flexure VSA.

Each of the columns in Table 5.1 describe a key functional characteristic that the VSA provides, and can be compared between each other. The VSA options are the stiffness ratio, constant power needs, free joint availability, range of motion, force feedback requirement, type of VSA, and time to change stiffness.

Table 5.1 shows that the Arched Flexure VSA provides the best VSA option currently available. It provides one of the largest stiffness ratios, provides 360 degrees of rotation, changes stiffness very quickly, provides a free joint, and is very compact. While current VSA options provide some of these capabilities, none of the current designs provide all of these capabilities. The goal of the Arched Flexure VSA was to create a VSA that exceeded current capabilities and that could replace current robot joints. In this regard, the Arched Flexure VSA design was successful.

Table 5.1: Summary of the VSA Options and Their Abilities

Name	**Constant Power Needed**	**Estimated Stiffness Ratio**	**Free Joint**	**Range of Motion**	**Force Feedback Required**	**Type**	**Full Range Variation Time**
Bidirectional Antagonistic VSA	Yes	37.5	No	203°	No	Antagonistic	0.014s
VSA Cube	Yes	4.637	No	120°	No	Antagonistic	0.32s
MACCEPA II	Yes	22	No	150°	No	Mechanical	2.6s
VSA-HD	No	22000	No	∞	Yes	Mechanical	0.4s
DLR FSJ	No	15.75	No	180°	No	Mechanical	0.33s
CompAct-VSA	No	50	Yes	120°	No	Mechanical	0.1s
vsaUT-II	No	70	Yes	120°	No	Mechanical	0.5s
VSJ-Leaf Springs	No	14.467	No	∞	No	Structure	0.2s
Arched Flexure VSA	No	55	Yes	∞	No	Structure	0.12s

5.2 Future Work

5.2.1 Design Modification

The Arched Flexure VSA design objectives should be modified by implementing a new optimization. The objective function, from Chapter 2, could be modified to include a compactness term, something like the following:

$$F(x) = \left(B * (X^2 + (R_1 + h_f/2)^2)^{(1/2)} - \log\left(\frac{k_{\max}}{k_{\min}}\right)\right) \tag{5.1}$$

This would take the location of the stiffness selection system into consideration, involving the design goal from Chapter 1. The only change this would add to the optimization code would be a new objective function that involves this, and new weight for the two portions of it. That makes this an easy change to make, if this approach is taken. This change incorporates the compactness of the design as a function of the flexure design. It also

removes the relative stiffness component from the design objectives, as experiments show that the relative stiffness was not met after the overall VSA was created. If this objective function is not used, the maximum stiffness ratio only optimization should be used. Since the results from Chapter Four show the VSA stiffness becomes nonlinear once the remainder of the components are applied, the flexure optimized for maximum stiffness range can be used instead of the manufactured flexure. This will decrease the low end stiffness, and increase the stiffness ratio even if the VSA is created at the same component stiffness level. The relative sensitivity is changed by the remainder of the VSA design too much to be included as a major component of the design, as changes in stiffness levels will change the relative sensitivity greatly.

5.2.2 Alternative Design Structure

Taking the current VSA design, several improvements can be made. In addition, the design can be modified to involve other design concepts. The most important change should be making the components of the VSA stiffer. This will increase the stiffness ratio of the VSA as a whole. To create additional stiffness multiple changes to the design should occur. First, higher ABEC rating bearings should be used, in addition to the chosen bearings having higher stiffness values. This will decrease positioning error, and also increase the stiffness of the joint. Second, any cantilevered components should be changed so they are no longer cantilevered. This can be done by adding a plate to the design, which will cause the design to be less compact. While it loses compactness, it also increase stiffness, and since the component stiffness is currently the major concern, the plate should be added to future designs. Third, the VSA should have an additional encoder added to the internal structure. The encoder should measure the overall joint angle. This will allow the joint angle to be incorporated into controllers, instead of only the motor joint position.

Finally, the VSA should be created out of nitinol, which is less stiff than titanium. Making the VSA out of a less stiff material will increase the relative stiffness of the

remainder of the VSA. In addition, the nitinol has additional properties that make it more desirable than the titanium. It operates as a superelastic material, meaning that instead of plastically deforming when too much force is applied, it instead undergoes a reversible change in phase and geometry for a large force range before plastically deforming. This allows it to handle larger accidental forces than the titanium could.

There are several design choices that could radically change the application of the VSA based on this design. The VSA can have offset stiffness selection systems for the two different directions of rotation. This would allow the VSA to have two different stiffness curves depending on which side is providing the stiffness. They would change at the same rate, but would have different stiffnesses at any given point, as they contact at different locations. The VSA would have the same stiffness ratio for one side, but the other side would have a lower stiffness ratio for the same motor angle, and would provide a free joint more often. This change can be created on the current VSA as well, just by assembling the stiffness selection systems at this offset. The two stiffness curves create directionally different stiffness for the VSA.

Taking this approach, the VSA could also have one stiffness selection set at a low or high stiffness value, and the other could change stiffness. This could have interesting applications in which one direction is required to always have high stiffness, and the other direction provides variable stiffness. Or if low stiffness is always required, where one always has low stiffness, and the other has variable. The current VSA could also have this change, if the gears connecting two of the connecting bars are removed and the bars are epoxied in place. There could be many different applications for this type of VSA involving human interaction. Perhaps a robot the requires repeated contact with a person could always have low stiffness on the side that contacts the person, and a variable stiffness in the other direction. This can be taken even further, and have one of the four contact selections set at a stiffness. This would allow one direction to have the full variable stiffness, and the other to have a composite stiffness of the set stiffness and the variable stiffness.

5.2.3 Additional Verification Strategy

The designed VSA can also be put in a three degree of freedom robot to perform planar tasks, which would allow the three stiffness values to be variable. This allows the robot to change it's stiffness to whatever values it requires, allowing it to assemble and perform many tasks. For example, turning a crank requires high stiffness tangential to the crank and low stiffness normal to it. By constantly changing it's stiffness to keep those requirements, a robot could turn the crank quickly while accommodating any small errors that might occur in robot position. It could also be applied to additional concepts like optimized assembly problems, just by changing the VSA to show those optimized values. In essence, the VSA becomes the tool to realize many manufacturing and assembly tasks expanding what robots can do.

BIBLIOGRAPHY

Albu-Schaffer, A., Eiberger, O., Grebenstein, M., Haddadin, S., Ott, C., Wimbock, T., Wolf, S., and Hirzinger, G. (2008). Soft robot: From torque feedback-controlled lightweight robot to intrinsically compliant systems. *IEEE Robotics & Automation Magazine*.

Carloni, R. and Marconi, L. (2012). Limit cycles and stiffness control with variable stiffness actuators. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Catalano, M. G., Grioli, G., Bonomo, F., Schiavi, R., and Bicchi, A. (2010). VSA-HD: From the enumeration analysis to the prototypical implementation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Catalano, M. G., Grioli, G., Garabini, M., Bonomo, F., Mancini, M., Tsagarakis, N., and Bicchi, A. (2011). VSA-cubebot: a modular variable stiffness platform for multiple degrees of freedom robots. In *IEEE International Conference on Robotics and Automation*.

Cherelle, P., Grosu, V., Beyl, P., Mathys, A., Ham, R. V., Damme, M. V., Vanderborght, B., and Lefeber, D. (2010). The MACCEPA actuation system as torque actuator in the gait rehabilitation robot ALTACRO. In *IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*.

Choi, J., Hong, S., Lee, W., and Kang, S. (2009). A variable stiffness joint using leaf springs for robot manipulators. In *IEEE International Conference on Robotics and Automation*.

Choi, J., Hong, S., Lee, W., Kang, S., and Kim, M. (2011). A robot joint with variable stiffness using leaf springs. *IEE Transactions on Robotics*, 37(2):229–238.

Groothuis, S., Rusticelli, G., Zucchelli, A., Stramigioli, S., and Carloni, R. (2012). The vsaUT-II: a novel rotational variable stiffness actuator. In *IEEE International Conference on Robotics and Automation*.

Ham, R. V., Sugar, T. G., Vanderborght, B., Hollander, K. W., and Lefeber, D. (2009). Compliant Actuator Design: Review of Actuators with Passive Adjustable Compliance/Controllable Stiffness for Robotic Applications. *IEEE Robotics & Automation Magazine*, 16(3):81–94.

Han, B., Zoppi, M., and Molfino, R. (2013). Variable impedance actuation using biphasic media. *Mechanism and Machine Theory*.

Petit, F., Chalon, M., Friedl, W., Grebenstein, M., Albu-Schaffer, A., and Hirzinger, G. (2010). Bidirectional antagonistic variable stiffness actuation: Analysis, design & implementation. In *IEEE International Conference on Robotics and Automation*.

Schiavi, R., Grioli, G., Sen, S., and Bicchi, A. (2008). VSA-II: a Novel Prototype of Variable Stiffness Actuator for Safe and Performing Robots Interacting with Humans. In *IEEE International Conference on Robotics and Automation*.

Tonietti, G., Schiavi, R., and Bicchi, A. (2005). Design and control of a variable stiffness actuator for safe and fast physical human/robot interaction. In *International Conference on Robotics and Automation*.

Tsagarakis, N. G., Sardellitti, I., and Caldwell, D. G. (2011). A new variable stiffness actuator (CompAct-VSA): Design and modelling. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Vanderborght, B., Ham, R. V., Lefeber, D., Sugar, T. G., and Hollander, K. W. (2009a). Comparison of mechanical design and energy consumption of adaptable, passive-compliant actuators. *The International Journal of Robotics Research*.

Vanderborght, B., Tsagarakis, N., Ham, R. V., Thorson, I., and Caldwell, D. G. (2011). MACCEPA 2.0: compliant actuator used for energy efficient hopping robot Chobino1D. *Autonomous Robotics*.

Vanderborght, B., Tsagarakis, N. G., Semini, C., Ham, R. V., and Caldwell, D. G. (2009b). MACCEPA 2.0: Adjustable compliant actuator with stiffening characteristics for energy efficient hopping. In *IEEE International Conference on Robotics and Automation*.

Visser, L. C., Carloni, R., and Stramigioli, S. (2011). Energy-efficient variable stiffness actuators. *IEEE Transactions on Robotics*, 27(5):865–875.

Wolf, S., Eiberger, O., and Hirzinger, G. (2011). The DLR FSJ: Energy based design of a variable stiffness joint. In *IEEE International Conference on Robotics and Automation*.

APPENDIX A

APPENDIX A: MATLAB TO ANSYS CODE

To properly perform the analysis, MatLab® was interfaced with ANSYS®. This interface combined the FEA abilities of ANSYS® with the optimization routines of MatLab®. ANSYS® provides a FEA program with the ability to be run in batch mode, allowing the program to just analyze the shape input through a text file. Batch mode turns off the GUI, and makes the program less responsive, but decreases the processing time by a significant amount of time, as well as allowing it to be run for multiple shapes. The genetic optimization routines in MatLab® will handle the optimization portion of the analysis, but cannot readily handle the FEA portion. Therefore a link between the two systems had to be created. This was done by sending text files to ANSYS® by using the system commands in MatLab®.

MatLab® will send ANSYS® the text file with the code to create the shape and the values for the variables used in the procedural generation of the flexure shape. ANSYS® then performs the FEA of the flexure shape described by the variables sent by MatLab®. ANSYS® outputs to text files the beam's shape and deformed shape for various force application locations. Once ANSYS® finishes the analysis of the flexure shape, it sends a command to MatLab®, which resumes the MatLab® portion of the code, parsing the data from the text files into the relevant variables. This code can be seen in Appendix []. Note that MatLab® is not run in parallel, because ANSYS® automatically runs in parallel. If MatLab® was run in parallel, it would cause the computer to error out, as ANSYS® would attempt to use processors that were dedicated to a parallel portion of MatLab®. This issue is why the MatLab® portion of the code is not set up to run in parallel.

Several key changes must be made to properly run the optimization and code. First, the text file must be in the same folder as the rest of the optimization files for the code to run

properly, and MatLab® and ANSYS® must be allowed to write to those files. The file locations in the text file and the m-file must be set to their current locations. Finally, the code for both the ANSYS® and MatLab® portions must be for the same locations.

A.1 Results Reliability Analysis

To understand the results and why they are valid, the mesh creation method must first be understood. Finite Element Analysis breaks up a given shape into a large number of linked tetrahedrons (meshing). The more refined the mesh, the more accurate the result of the analysis typically are, until numerical error dominates. In this case, ANSYS® procedurally creates a series of volumes and then combines these volumes to create the flexure shape. In addition, the stiffness analysis locations are required to be along the initial volume edges, because ANSYS® requires it for proper force application. In addition, the keypoints and lines used to create the volumes must be maintained so the forces can be applied to them. Therefore, ANSYS® must create the mesh while maintaining those keypoints, lines, and volumes. ANSYS® requires a minimum number of elements for any given volume or line.

Therefore, the smaller volumes at the low stiffness application locations have meshes that closely match each other at different mesh refinement levels, as the minimum element requirement dominates. The larger volumes don't begin to approach the element density of the smaller volumes until larger mesh refinement levels. See Figure A.1 for an image of flexure with mesh superimposed. Figure A.1 shows the mesh matching at various force application points due to the larger mesh refinement levels. Several flexure shapes were then analyzed for stiffness at the different refinement levels. Each of the resulting stiffness plots were then compared to confirm that they followed the same pattern, that of the lower stiffness matching and the higher stiffness decreasing. Since the pattern was maintained, the optimization at lower refinement levels can still be used.

The individual element displacements were also examined to confirm validity of the model. See Figure A.2 for an image of the displacements of elements when subjected to a

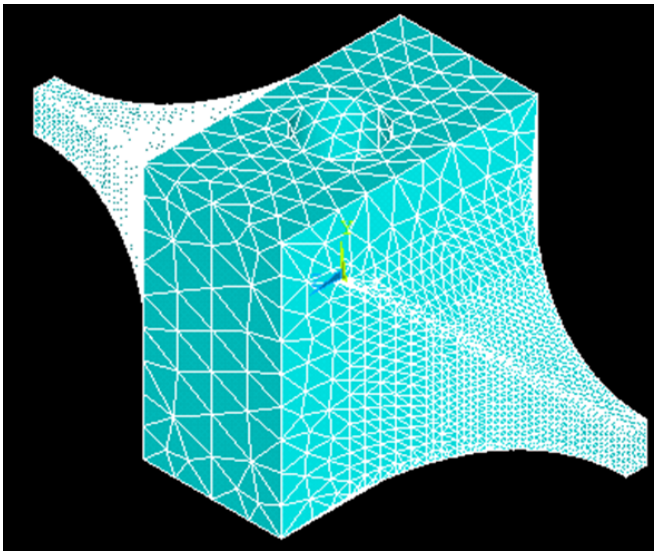


Figure A.1: Flexure with Mesh Superimposed Throughout the Volume

force on the end of the flexure. Figure A.2 shows the displacements given a force at the end of the flexure, or when the application angle is zero. The displacements makes sense, as the elements all show a rotation about the center point while displacing greater amounts the further away from the center point. In essence, the elements all displace roughly perpendicular to the line connecting the element to the center axis, which is expected.

Next, the results from a middle application point was checked for validity. See Figure A.3 for the image of the displacements of elements close to the force application point when subjected to a force at an angle in the middle of the flexure. Figure A.3 shows the displacements of elements close the the force application point. Note that the elements further away from the force application point react in a similar way to the elements seen in Figure A.2. The elements at the force application point do not react in a similar way, because those elements see both the rotation about the center point as well as the compression of those elements due to the force application. In this case, the combination of effects causes the element to react in the direction indicated, not in the direct direction of the force.

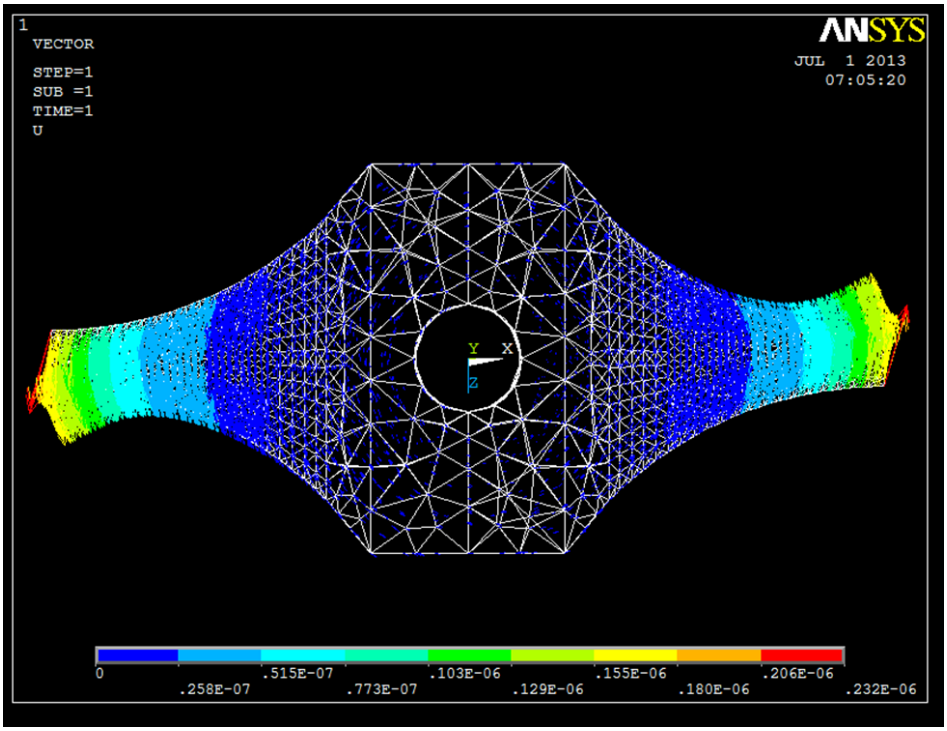


Figure A.2: Element Displacements Given a Force at the End of the Flexure

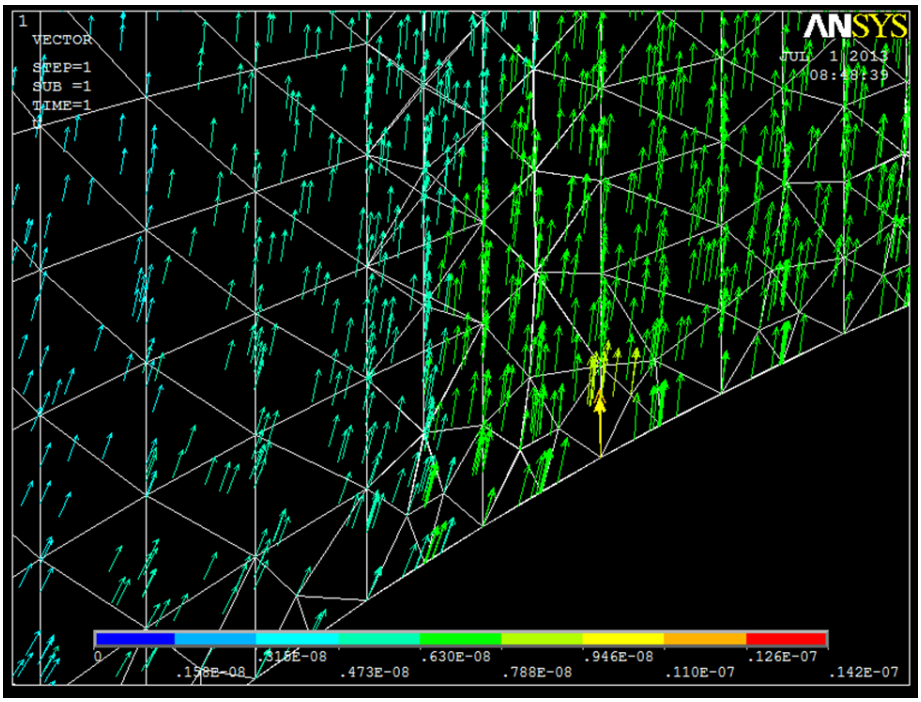


Figure A.3: Element Displacements Given a Force at an Angle in the Middle of the Flexure

A.2 Generic Matlab Code

Below is the generic MatLab code that is used for all of the Optimizations.

The overall m-file that handles the Optimizations. Note that the combinedtrial function is changed to represent the 5, 9, or 21 location to match the optimization being run. The MatLab® code and ANSYS® code for the 9 and 21 location evaluations can be accessed at Marquette University in the MECH Sys Lab.

```
%Dan Garces
%4/1/13
global R2 w1 w2
w1= 10;  %Weight of Ratio
w2= 1;  %Weight of Straightness
R2=.15;
count=0;

A=[-1,0,0,1,0,0,0;0,-1,0,0,1,0,0;0,-1,0,0,0,0,1];

b=[-10*pi/180;-0.1;-0.1];

lb=[22.5/180*pi;1.01/R2;0.05/R2;15/180*pi;0.75/R2;0.05/R2;0.75/R2];

ub=[85/180*pi;5/R2;.25/R2;45/180*pi;2/R2;0.25/R2;2/R2];
ubp=ub';
lbp=lb';
options=gaoptimset('PopInitRange', [lbp;ubp], 'PopulationSize', ...
    ↪ [20], 'MigrationDirection', 'both', 'MigrationInterval', 5, ...
    ↪ 'MigrationFraction', 0.2, 'PlotFcns', ...
    ↪ {@gaplotbestf,@gaplotstopping}, 'Generations', 20, ...
    ↪ 'InitialPopulation', ...
    ↪ [.9599,7.7333,1.1896,.7854,7.1333,.5292,6.5; ...
    ↪ .9599,8.,1.1896,.7854,7.3333,.5292,6.7;], 'UseParallel', ...
    ↪ 'Always');
[numberss, valuess, exitflagss, outputss, populationss, scoresss] ...
    ↪ = ga(@combinedtrial,7,A,b,[],[],lb,ub,@confun,[],options)
```

The constraining function used by the overall m-file.

```
function [c,ceq] = confun(x)
%Nonlinear Constraints Function for Optimization
global R2
R3=R2*x(7);
wf=R2*x(6);
theta1=x(1);
hf=R2*x(3);
R1=R2*x(5);
```

```
X=R2*x(2);
theta2=x(4)
A=-R3-wf/2+R3*cos(theta1)+R2*1.5;
B=-R1-hf/2+R1*cos(theta1)+R2*1.5;
C=-X+R2*1.5+R3*sin(theta1);
D=-X+R2*1.5+R1*sin(theta1);
E=-tan(X/(hf/2+R1))+theta2+0.05;
c=[A;B;C;D;E];
ceq=[];
end
```

A.3 5 Location Optimization

Below is the MatLab® code for the 5 location evaluation.

```
global w1 w2 R2
tic
a1=x(1)
a2=x(2)
a3=x(3)
a4=x(4)
a5=x(5)
a6=x(6)
a7=x(7)
%Create the inputs for the trial run
theta1=a1;
theta2=a4;
X=a2*R2;
hf=a3*R2;
R1=a5*R2;
wf=a6*R2;
R3=a7*R2;
P=R2*2000/3;
if theta1<theta2
    overalls=1000000000;
elseif X-R1*sin(theta1)<0
    overalls=1000000000;
elseif X-R3*sin(theta1)<0
    overalls=1000000000;
end
fclose('all');
delete('TRIALS3.txt');
delete('TRIALS4.txt');
delete('TRIALS5.txt');
delete('TRIALS6.txt');
delete('TRIALS7.txt');
delete('TRIALS8.txt');
delete('TRIALS9.txt');

D3=fopen('TRIALS3.txt','w');
D3=fopen('TRIALS4.txt','w');
```

```
D3=fopen('TRIALS5.txt','w');
D3=fopen('TRIALS6.txt','w');
D3=fopen('TRIALS7.txt','w');
D3=fopen('TRIALS8.txt','w');
D3=fopen('TRIALS9.txt','w');
fclose('all');
%setenv('ANS_CONSEC','YES')
%Create the string to open ansys
ProgStr = 'C:\"Program Files"\"ANSYS Inc"\v140\ansys\bin\WINX64\ANSYS140.exe ';
OptStr  = ' -p aa_t_i -b -t6 -i Trials111.txt -o Trials2.txt';
%OptStr  = ' -p aa_t_i -b -t6 -i BasicCommandsTest.txt -o Trials2.txt';
CmdStr  = [ProgStr '-R1 ' num2str(R1) ' -Force ' num2str(P) ' -R2 ' num2str(R2) ' -hf ' num2str(hf) ' -xlength ' num2str(X) ' -theta1 ' num2str(theta1) ' -theta2 ' num2str(theta2) ' -wf ' num2str(wf) ' -R3 ' num2str(R3) OptStr];

FAIL=system(CmdStr);

clear Xstore
clear ELEstore
clear Nodestore
Xstore=[];
Nodestore=[];
ELEstore=[];
% Open output file for examination
D=fopen('TRIALS3.txt','r');
counter=0;
for i=1:50
    line=fgetl(D);
    counter=counter+1;
    if length(line)==32;
        line=fgetl(D);
        break
    else
    end
end
counter2=1;
for i=1:50000
    Xstore=fgetl(D);
    if length(Xstore)==15
        break
    elseif length(Xstore)≤32;
        for i=1:12
            Xstore=fgetl(D);
        end
    else
        Xstore2=str2num(Xstore(1:8));
        Xstore3=str2num(Xstore(9:21));
        Xstore4=str2num(Xstore(22:33));
        Xstore5(counter2,:)=[Xstore2,Xstore3,Xstore4];
        counter2=counter2+1;
    end
```

```
end

%Option2
D=fopen('TRIALS6.txt','r');
counter=0;
for i=1:50
    line=fgetl(D);
    counter=counter+1;
    if length(line)==32;
        line=fgetl(D);
        break
    else
    end
end
counter2=1;
for i=1:50000
    Xstore=fgetl(D);
    if length(Xstore)==15
        break
    elseif length(Xstore)≤32;
        for i=1:12
            Xstore=fgetl(D);
        end
    else
        Xstore2=str2num(Xstore(1:8));
        Xstore3=str2num(Xstore(9:21));
        Xstore4=str2num(Xstore(22:33));
        Xstore6(counter2,:)=[Xstore2,Xstore3,Xstore4];
        counter2=counter2+1;
    end
end

%Option3
D=fopen('TRIALS7.txt','r');
counter=0;
for i=1:50
    line=fgetl(D);
    counter=counter+1;
    if length(line)==32;
        line=fgetl(D);
        break
    else
    end
end
counter2=1;
for i=1:50000
    Xstore=fgetl(D);
    if length(Xstore)==15
        break
    elseif length(Xstore)≤32;
        for i=1:12
            Xstore=fgetl(D);
        end
    else
```

```
        Xstore2=str2num(Xstore(1:8));
        Xstore3=str2num(Xstore(9:21));
        Xstore4=str2num(Xstore(22:33));
        Xstore7(counter2,:)=[Xstore2,Xstore3,Xstore4];
        counter2=counter2+1;
    end
end

%Option4
D=fopen('TRIALS8.txt','r');
counter=0;
for i=1:50
    line=fgetl(D);
    counter=counter+1;
    if length(line)==32;
        line=fgetl(D);
        break
    else
    end
end
counter2=1;
for i=1:50000
    Xstore=fgetl(D);
    if length(Xstore)==15
        break
    elseif length(Xstore)≤32;
        for i=1:12
            Xstore=fgetl(D);
        end
    else
        Xstore2=str2num(Xstore(1:8));
        Xstore3=str2num(Xstore(9:21));
        Xstore4=str2num(Xstore(22:33));
        Xstore8(counter2,:)=[Xstore2,Xstore3,Xstore4];
        counter2=counter2+1;
    end
end

%Option5
D=fopen('TRIALS9.txt','r');
counter=0;
for i=1:50
    line=fgetl(D);
    counter=counter+1;
    if length(line)==32;
        line=fgetl(D);
        break
    else
    end
end
counter2=1;
for i=1:50000
    Xstore=fgetl(D);
    if length(Xstore)==15
```

```
194         break
195     elseif length(Xstore)≤32;
196         for i=1:12
197             Xstore=fgetl(D);
198         end
199     else
200         Xstore2=str2num(Xstore(1:8));
201         Xstore3=str2num(Xstore(9:21));
202         Xstore4=str2num(Xstore(22:33));
203         Xstore9(counter2,:)=[Xstore2,Xstore3,Xstore4];
204         counter2=counter2+1;
205     end
206 end
207
208
209
210 Q=fopen('TRIALS4.txt','r');
211 counter=1;
212 for i=1:50000;
213     nodes2=fgetl(Q);
214     if nodes2==-1
215         break
216     else
217         Nodestore1=str2num(nodes2(1:8));
218         if length(nodes2)==8;
219             Nodestore2=0;
220             Nodestore3=0;
221             Nodestore4=0;
222         elseif length(nodes2)==28;
223             Nodestore2=str2num(nodes2(9:28));
224             Nodestore3=0;
225             Nodestore4=0;
226         elseif length(nodes2)==48
227             Nodestore2=str2num(nodes2(9:28));
228             Nodestore3=str2num(nodes2(29:48));
229             Nodestore4=0;
230         else
231             Nodestore2=str2num(nodes2(9:28));
232             Nodestore3=str2num(nodes2(29:48));
233             Nodestore4=str2num(nodes2(49:length(nodes2)));
234         end
235         Nodestore(counter,:)= ...
                ↪ [Nodestore1,Nodestore2,Nodestore3,Nodestore4];
236         counter=counter+1;
237     end
238 end
239
240 QQ=fopen('TRIALS5.txt','r');
241 counter=1;
242 for i=1:50000
243         nodes2=fgetl(QQ);
244     if nodes2==-1
245         break
246     else
```

```
247          Nodestore1=str2num(nodes2(1:6));
248          Nodestore2=str2num(nodes2(7:12));
249          Nodestore3=str2num(nodes2(13:18));
250          Nodestore4=str2num(nodes2(19:24));
251          Nodestore5=str2num(nodes2(25:30));
252          Nodestore6=str2num(nodes2(31:36));
253          Nodestore7=str2num(nodes2(37:42));
254          Nodestore8=str2num(nodes2(43:48));
255          nodes2=fgetl(QQ);
256          Nodestore9=str2num(nodes2(1:6));
257          Nodestore10=str2num(nodes2(7:12));
258          ELEstore(counter,:)= ...
                 ↪ [Nodestore1,Nodestore2,Nodestore3,Nodestore4, ...
                 ↪ Nodestore5,Nodestore6,Nodestore7,Nodestore8, ...
                 ↪ Nodestore9,Nodestore10];
259          counter=counter+1;
260      end
261  end
262  fclose('all');
263  % [d1,d2]=sort(Nodestore(:,2));
264  % B=Nodestore(d2,:);
265  counter1=1;
266  counter2=1;
267  counter3=1;
268  counter4=1;
269  counter5=1;
270  ratio=1*10^-5;
271  for i=1:length(Nodestore);
272      if Nodestore(i,2)≥(-X-ratio) & Nodestore(i,2)≤(-X+ratio);
273          if Nodestore(i,4)≥(-hf/2-ratio) & ...
                 ↪ Nodestore(i,4)≤(-hf/2+ratio);
274              storenode1(counter1,:)=Nodestore(i,:);
275              counter1=counter1+1;
276          end
277      elseif Nodestore(i,2)≥(-X+R1*sin(theta2/2)-ratio) & ...
             ↪ Nodestore(i,2)≤(-X+R1*sin(theta2/2)+ratio);
278          if Nodestore(i,4)≥ (-R1-hf/2+R1*cos(theta2/2)-ratio) & ...
                 ↪ Nodestore(i,4)≤(-R1-hf/2+R1*cos(theta2/2)+ratio);
279              storenode3(counter2,:)=Nodestore(i,:);
280              counter2=counter2+1;
281          end
282      elseif Nodestore(i,2)≥(-X+R1*sin(theta2)-ratio) & ...
             ↪ Nodestore(i,2)≤(-X+R1*sin(theta2)+ratio);
283          if Nodestore(i,4)≥(-R1-hf/2+R1*cos(theta2)-ratio) & ...
                 ↪ Nodestore(i,4)≤(-R1-hf/2+R1*cos(theta2)+ratio);
284              storenode5(counter3,:)=Nodestore(i,:);
285              counter3=counter3+1;
286          end
287      elseif Nodestore(i,2)≥(-X+R1*sin(theta2/4)-ratio) & ...
             ↪ Nodestore(i,2)≤(-X+R1*sin(theta2/4)+ratio);
288          if Nodestore(i,4)≥(-R1-hf/2+R1*cos(theta2/4)-ratio) & ...
                 ↪ Nodestore(i,4)≤(-R1-hf/2+R1*cos(theta2/4)+ratio);
289              storenode2(counter4,:)=Nodestore(i,:);
290              counter4=counter4+1;
```

```
291         end
292     elseif Nodestore(i,2)≥(-X+R1*sin(theta2/1.5)-ratio) & ...
            ↪ Nodestore(i,2)≤(-X+R1*sin(theta2/1.5)+ratio);
293         if Nodestore(i,4)≥(-R1-hf/2+R1*cos(theta2/1.5)-ratio) & ...
                ↪ Nodestore(i,4)≤(-R1-hf/2+R1*cos(theta2/1.5)+ratio);
294             storenode4(counter5,:)=Nodestore(i,:);
295             counter5=counter5+1;
296         end
297     else
298     end
299 end
300 [d11,d12]=sort(storenode1(:,3));
301 [d21,d22]=sort(storenode2(:,3));
302 [d31,d32]=sort(storenode3(:,3));
303 [d41,d42]=sort(storenode4(:,3));
304 [d51,d52]=sort(storenode5(:,3));
305 B1=storenode1(d12,:);
306 B2=storenode2(d22,:);
307 B3=storenode3(d32,:);
308 B4=storenode4(d42,:);
309 B5=storenode5(d52,:);
310
311 storage11=[];
312 storage22=[];
313 storage33=[];
314 storage44=[];
315 storage55=[];
316 storage11r=[];
317 storage22r=[];
318 storage33r=[];
319 storage44r=[];
320 storage55r=[];
321 counter1=1;
322 counter2=1;
323 counter3=1;
324 counter4=1;
325 counter5=1;
326 for i=1:length(ELEstore)
327     for k=1:min(size(ELEstore))
328         for j=1:length(B1(:,1))
329             if ELEstore(i,k)==B1(j,1)
330                 storage11(counter1,1)=i;
331                 counter1=counter1+1;
332             end
333         end
334         for j=1:length(B2(:,1))
335             if ELEstore(i,k)==B2(j,1)
336                 storage22(counter2,1)=i;
337                 counter2=counter2+1;
338             end
339         end
340
341         for j=1:length(B3(:,1))
342             if ELEstore(i,k)==B3(j,1)
```

```
343                 storage33(counter3,1)=i;
344                 counter3=counter3+1;
345             end
346         end
347
348         for j=1:length(B4(:,1))
349             if ELEstore(i,k)==B4(j,1)
350                 storage44(counter4,1)=i;
351                 counter4=counter4+1;
352             end
353         end
354
355         for j=1:length(B5(:,1))
356             if ELEstore(i,k)==B5(j,1)
357                 storage55(counter5,1)=i;
358                 counter5=counter5+1;
359             end
360         end
361
362     end
363 end
364 counter1=1;
365 counter2=1;
366 counter3=1;
367 counter4=1;
368 counter5=1;
369 for j=1:length(storage11)
370     for jj=1:length(storage11)
371         if j≥jj
372             continue
373         end
374         if storage11(j,1)==storage11(jj,1)
375             result=storage11(j,1);
376             if counter1>1;
377                 if storage11r(counter1-1,1)==result;
378                     continue
379                 else
380                     storage11r(counter1,1)=result;
381                     counter1=counter1+1;
382                 end
383             else
384                 storage11r(counter1,1)=result;
385                 counter1=counter1+1;
386             end
387         end
388     end
389 end
390 for j=1:length(storage22)
391     for jj=1:length(storage22)
392         if j≥jj
393             continue
394         end
395         if storage22(j,1)==storage22(jj,1)
396             result=storage22(j,1);
```

```
            if counter2>1;
                if storage22r(counter2-1,1)==result;
                    continue
                else
                    storage22r(counter2,1)=result;
                    counter2=counter2+1;
                end
            else
                storage22r(counter2,1)=result;
                counter2=counter2+1;
            end
        end
    end
end
for j=1:length(storage33)
    for jj=1:length(storage33)
        if j≥jj
            continue
        end
        if storage33(j,1)==storage33(jj,1)
            result=storage33(j,1);
            if counter3>1;
                if storage33r(counter3-1,1)==result;
                    continue
                else
                    storage33r(counter3,1)=result;
                    counter3=counter3+1;
                end
            else
                storage33r(counter3,1)=result;
                counter3=counter3+1;
            end
        end
    end
end
for j=1:length(storage44)
    for jj=1:length(storage44)
        if j≥jj
            continue
        end
        if storage44(j,1)==storage44(jj,1)
            result=storage44(j,1);
            if counter4>1;
                if storage44r(counter4-1,1)==result;
                    continue
                else
                    storage44r(counter4,1)=result;
                    counter4=counter4+1;
                end
            else
                storage44r(counter4,1)=result;
                counter4=counter4+1;
            end
        end
```

```
451     end
452 end
453 for j=1:length(storage55)
454     for jj=1:length(storage55)
455         if j≥jj
456             continue
457         end
458         if storage55(j,1)==storage55(jj,1)
459             result=storage55(j,1);
460             if counter5>1;
461                 if storage55r(counter5-1,1)==result;
462                     continue
463                 else
464                     storage55r(counter5,1)=result;
465                     counter5=counter5+1;
466                 end
467             else
468                 storage55r(counter5,1)=result;
469                 counter5=counter5+1;
470             end
471         end
472     end
473 end
474 counter1=1;
475 counter2=1;
476 counter3=1;
477 counter4=1;
478 counter5=1;
479
480
481
482 %Portion takes the 5 elements found above and determines their ...
    ↪ dispacements
483 %in the relevant cases
484
485 for i=1:length(storage11r);
486     result1(i,:)=Xstore5(storage11r(i),:);
487 end
488 for i=1:length(storage22r);
489     result2(i,:)=Xstore6(storage22r(i),:);
490 end
491 for i=1:length(storage33r);
492     result3(i,:)=Xstore7(storage33r(i),:);
493 end
494 for i=1:length(storage44r);
495     result4(i,:)=Xstore8(storage44r(i),:);
496 end
497 for i=1:length(storage55r);
498     result5(i,:)=Xstore9(storage55r(i),:);
499 end
500 if min(size(result1))==1;
501     result11=result1;
502 else
503     result11=mean(result1);
```

```
end
if min(size(result2))==1;
    result22=result2;
else
    result22=mean(result2);
end
if min(size(result3))==1;
    result33=result3;
else
    result33=mean(result3);
end
if min(size(result4))==1;
    result44=result4;
else
    result44=mean(result4);
end
if min(size(result5))==1;
    result55=result5;
else
    result55=mean(result5);
end

results1=P/atan(abs(result11(1,3)/(X-result11(1,2))))
results2=P/atan(abs(result22(1,3)/(X-result22(1,2))))
results3=P/atan(abs(result33(1,3)/(X-result33(1,2))))
results4=P/atan(abs(result44(1,3)/(X-result44(1,2))))
results5=P/atan(abs(result55(1,3)/(X-result55(1,2))))
resulters=[results1,results2,results3,results4,results5];
%Portion takes the angular displacement of the 5 cases, and ...
    ↪ analyzes them for the
%function results, including the weighting function.
resultlog(1)=abs((log(results1)-log(results2))/(theta2/4));
resultlog(2)=abs((log(results2)-log(results3))/(theta2/2-theta2/4));
resultlog(3)=abs((log(results3)-log(results4))/(theta2/1.5-theta2/2));
resultlog(4)=abs((log(results4)-log(results5))/(theta2-theta2/1.5));
resultlog15=abs((log(results5)-log(results1))/(theta2));
difference=0
for i=1:4;
    difference=difference+abs(resultlog15-resultlog(i))
end

maxtomin=max(resulters)-min(resulters)

overalls=-(difference*w1+maxtomin*w2)

fclose('all');   %Closes the file, so it can be redone in the ...
    ↪ next iteration
toc
semilogy([0,theta2/4,theta2/2,theta2/1.5,theta2],resulters)
grid on
ylabel('Stiffness (Nm/rad')
xlabel('Angle (rad)')
```

Below is the ANSYS® code for the 5 location evaluation.

```
/BATCH      !Tells ANSYS this is a batch file,  fails without
!Dan Garces
!03/06/13
!Input File to handle optimization
!Inputs are:  R1, R2, hf, xlength, theta1, theta2, wf, Force

/PREP7
ET, 1, SOLID187
MP, EX, 1, 2.0E11
MP, PRXY, 1, 0.3
DOF, UX, UY, UZ
!Define Keypoints: k, n, x, y, z
K, 1, -xlength, -wf/2, -hf/2
K, 2, -xlength, -wf/2, hf/2
K, 3, -xlength, wf/2, hf/2
K, 4, -xlength, wf/2, -hf/2
K, 5, -xlength+R1*sin(theta1), -R1-wf/2+R1*cos(theta1), ...
    ↪ -R1-hf/2+R1*cos(theta1)
K, 6, -xlength+R1*sin(theta1), -R1-wf/2+R1*cos(theta1), ...
    ↪ R1+hf/2-R1*cos(theta1)
K, 7, -xlength+R1*sin(theta1), R1+wf/2-R1*cos(theta1), ...
    ↪ R1+hf/2-R1*cos(theta1)
K, 8, -xlength+R1*sin(theta1), R1+wf/2-R1*cos(theta1), ...
    ↪ -R1-hf/2+R1*cos(theta1)
K, 9, 0, -R1-wf/2+R1*cos(theta1), -R1-hf/2+R1*cos(theta1)
K, 10, 0, -R1-wf/2+R1*cos(theta1), R1+hf/2-R1*cos(theta1)
K, 11, 0, R1+wf/2-R1*cos(theta1), R1+hf/2-R1*cos(theta1)
K, 12, 0, R1+wf/2-R1*cos(theta1), -R1-hf/2+R1*cos(theta1)
K, 13, xlength-R1*sin(theta1), -R1-wf/2+R1*cos(theta1), ...
    ↪ -R1-hf/2+R1*cos(theta1)
K, 14, xlength-R1*sin(theta1), -R1-wf/2+R1*cos(theta1), ...
    ↪ R1+hf/2-R1*cos(theta1)
K, 15, xlength-R1*sin(theta1), R1+wf/2-R1*cos(theta1), ...
    ↪ R1+hf/2-R1*cos(theta1)
K, 16, xlength-R1*sin(theta1), R1+wf/2-R1*cos(theta1), ...
    ↪ -R1-hf/2+R1*cos(theta1)
K, 17, xlength, -wf/2, -hf/2
K, 18, xlength, -wf/2, hf/2
K, 19, xlength, wf/2, hf/2
K, 20, xlength, wf/2, -hf/2
K, 21, -xlength+R1*sin(theta2/2), -R1-wf/2+R1*cos(theta2/2), ...
    ↪ -R1-hf/2+R1*cos(theta2/2)
K, 22, -xlength+R1*sin(theta2/2), -R1-wf/2+R1*cos(theta2/2), ...
    ↪ R1+hf/2-R1*cos(theta2/2)
K, 23, -xlength+R1*sin(theta2/2), R1+wf/2-R1*cos(theta2/2), ...
    ↪ R1+hf/2-R1*cos(theta2/2)
K, 24, -xlength+R1*sin(theta2/2), R1+wf/2-R1*cos(theta2/2), ...
    ↪ -R1-hf/2+R1*cos(theta2/2)
K, 25, -xlength+R1*sin(theta2), -R1-wf/2+R1*cos(theta2), ...
    ↪ -R1-hf/2+R1*cos(theta2)
K, 26, -xlength+R1*sin(theta2), -R1-wf/2+R1*cos(theta2), ...
    ↪ R1+hf/2-R1*cos(theta2)
```

```
39 K, 27, -xlength+R1*sin(theta2), R1+wf/2-R1*cos(theta2), ...
       ↪ R1+hf/2-R1*cos(theta2)
40 K, 28, -xlength+R1*sin(theta2), R1+wf/2-R1*cos(theta2), ...
       ↪ -R1-hf/2+R1*cos(theta2)
41 K, 29, xlength-R1*sin(theta2), -R1-wf/2+R1*cos(theta2), ...
       ↪ -R1-hf/2+R1*cos(theta2)
42 K, 30, xlength-R1*sin(theta2), -R1-wf/2+R1*cos(theta2), ...
       ↪ R1+hf/2-R1*cos(theta2)
43 K, 31, xlength-R1*sin(theta2), R1+wf/2-R1*cos(theta2), ...
       ↪ R1+hf/2-R1*cos(theta2)
44 K, 32, xlength-R1*sin(theta2), R1+wf/2-R1*cos(theta2), ...
       ↪ -R1-hf/2+R1*cos(theta2)
45 K, 33, xlength-R1*sin(theta2/2), -R1-wf/2+R1*cos(theta2/2), ...
       ↪ -R1-hf/2+R1*cos(theta2/2)
46 K, 34, xlength-R1*sin(theta2/2), -R1-wf/2+R1*cos(theta2/2), ...
       ↪ R1+hf/2-R1*cos(theta2/2)
47 K, 35, xlength-R1*sin(theta2/2), R1+wf/2-R1*cos(theta2/2), ...
       ↪ R1+hf/2-R1*cos(theta2/2)
48 K, 36, xlength-R1*sin(theta2/2), R1+wf/2-R1*cos(theta2/2), ...
       ↪ -R1-hf/2+R1*cos(theta2/2)
49 K, 37, 0, 0, 0
50 K, 38, -xlength, -wf/4, -hf/2
51 K, 39, -xlength, 0, -hf/2
52 K, 40, -xlength, wf/4, -hf/2
53 K, 41, -xlength+R1*sin(theta2/2), (-R1-wf/2+R1*cos(theta2/2))/2, ...
       ↪ -R1-hf/2+R1*cos(theta2/2)
54 K, 42, -xlength+R1*sin(theta2/2), 0, -R1-hf/2+R1*cos(theta2/2)
55 K, 43, -xlength+R1*sin(theta2/2), (R1+wf/2-R1*cos(theta2/2))/2, ...
       ↪ -R1-hf/2+R1*cos(theta2/2)
56 K, 44, -xlength+R1*sin(theta2), (-R1-wf/2+R1*cos(theta2))/2, ...
       ↪ -R1-hf/2+R1*cos(theta2)
57 K, 45, -xlength+R1*sin(theta2), 0, -R1-hf/2+R1*cos(theta2)
58 K, 46, -xlength+R1*sin(theta2), (R1+wf/2-R1*cos(theta2))/2, ...
       ↪ -R1-hf/2+R1*cos(theta2)
59 K, 47, xlength-R1*sin(theta2), -(R1+wf/2-R1*cos(theta2))/2, ...
       ↪ R1+hf/2-R1*cos(theta2)
60 K, 48, xlength-R1*sin(theta2), 0, R1+hf/2-R1*cos(theta2)
61 K, 49, xlength-R1*sin(theta2), (R1+wf/2-R1*cos(theta2))/2, ...
       ↪ R1+hf/2-R1*cos(theta2)
62 K, 50, xlength-R1*sin(theta2/2), (-R1-wf/2+R1*cos(theta2/2))/2, ...
       ↪ R1+hf/2-R1*cos(theta2/2)
63 K, 51, xlength-R1*sin(theta2/2), 0, R1+hf/2-R1*cos(theta2/2)
64 K, 52, xlength-R1*sin(theta2/2), (R1+wf/2-R1*cos(theta2/2))/2, ...
       ↪ R1+hf/2-R1*cos(theta2/2)
65 K, 53, xlength, -wf/4, hf/2
66 K, 54, xlength, 0, hf/2
67 K, 55, xlength, wf/4, hf/2
68 K, 56, -xlength+R1*sin(theta2/4), -R1-wf/2+R1*cos(theta2/4), ...
       ↪ -R1-hf/2+R1*cos(theta2/4)
69 K, 57, -xlength+R1*sin(theta2/4), -R1-wf/2+R1*cos(theta2/4), ...
       ↪ R1+hf/2-R1*cos(theta2/4)
70 K, 58, -xlength+R1*sin(theta2/4), R1+wf/2-R1*cos(theta2/4), ...
       ↪ R1+hf/2-R1*cos(theta2/4)
```

```
71 K, 59, -xlength+R1*sin(theta2/4), R1+wf/2-R1*cos(theta2/4), ...
       ↪ -R1-hf/2+R1*cos(theta2/4)
72 K, 60, -xlength+R1*sin(theta2/1.5), -R1-wf/2+R1*cos(theta2/1.5), ...
       ↪ -R1-hf/2+R1*cos(theta2/1.5)
73 K, 61, -xlength+R1*sin(theta2/1.5), -R1-wf/2+R1*cos(theta2/1.5), ...
       ↪ R1+hf/2-R1*cos(theta2/1.5)
74 K, 62, -xlength+R1*sin(theta2/1.5), R1+wf/2-R1*cos(theta2/1.5), ...
       ↪ R1+hf/2-R1*cos(theta2/1.5)
75 K, 63, -xlength+R1*sin(theta2/1.5), R1+wf/2-R1*cos(theta2/1.5), ...
       ↪ -R1-hf/2+R1*cos(theta2/1.5)
76 K, 64, -xlength+R1*sin((theta1-theta2)/2+theta2), ...
       ↪ -R1-wf/2+R1*cos((theta1-theta2)/2+theta2), ...
       ↪ -R1-hf/2+R1*cos((theta1-theta2)/2+theta2)
77 K, 65, -xlength+R1*sin((theta1-theta2)/2+theta2), ...
       ↪ -R1-wf/2+R1*cos((theta1-theta2)/2+theta2), ...
       ↪ R1+hf/2-R1*cos((theta1-theta2)/2+theta2)
78 K, 66, -xlength+R1*sin((theta1-theta2)/2+theta2), ...
       ↪ R1+wf/2-R1*cos((theta1-theta2)/2+theta2), ...
       ↪ R1+hf/2-R1*cos((theta1-theta2)/2+theta2)
79 K, 67, -xlength+R1*sin((theta1-theta2)/2+theta2), ...
       ↪ R1+wf/2-R1*cos((theta1-theta2)/2+theta2), ...
       ↪ -R1-hf/2+R1*cos((theta1-theta2)/2+theta2)
80 K, 68, xlength-R1*sin((theta1-theta2)/2+theta2), ...
       ↪ -R1-wf/2+R1*cos((theta1-theta2)/2+theta2), ...
       ↪ -R1-hf/2+R1*cos((theta1-theta2)/2+theta2)
81 K, 69, xlength-R1*sin((theta1-theta2)/2+theta2), ...
       ↪ -R1-wf/2+R1*cos((theta1-theta2)/2+theta2), ...
       ↪ R1+hf/2-R1*cos((theta1-theta2)/2+theta2)
82 K, 70, xlength-R1*sin((theta1-theta2)/2+theta2), ...
       ↪ R1+wf/2-R1*cos((theta1-theta2)/2+theta2), ...
       ↪ R1+hf/2-R1*cos((theta1-theta2)/2+theta2)
83 K, 71, xlength-R1*sin((theta1-theta2)/2+theta2), ...
       ↪ R1+wf/2-R1*cos((theta1-theta2)/2+theta2), ...
       ↪ -R1-hf/2+R1*cos((theta1-theta2)/2+theta2)
84 K, 72, xlength-R1*sin(theta2/1.5), -R1-wf/2+R1*cos(theta2/1.5), ...
       ↪ -R1-hf/2+R1*cos(theta2/1.5)
85 K, 73, xlength-R1*sin(theta2/1.5), -R1-wf/2+R1*cos(theta2/1.5), ...
       ↪ R1+hf/2-R1*cos(theta2/1.5)
86 K, 74, xlength-R1*sin(theta2/1.5), R1+wf/2-R1*cos(theta2/1.5), ...
       ↪ R1+hf/2-R1*cos(theta2/1.5)
87 K, 75, xlength-R1*sin(theta2/1.5), R1+wf/2-R1*cos(theta2/1.5), ...
       ↪ -R1-hf/2+R1*cos(theta2/1.5)
88 K, 76, xlength-R1*sin(theta2/4), -R1-wf/2+R1*cos(theta2/4), ...
       ↪ -R1-hf/2+R1*cos(theta2/4)
89 K, 77, xlength-R1*sin(theta2/4), -R1-wf/2+R1*cos(theta2/4), ...
       ↪ R1+hf/2-R1*cos(theta2/4)
90 K, 78, xlength-R1*sin(theta2/4), R1+wf/2-R1*cos(theta2/4), ...
       ↪ R1+hf/2-R1*cos(theta2/4)
91 K, 79, xlength-R1*sin(theta2/4), R1+wf/2-R1*cos(theta2/4), ...
       ↪ -R1-hf/2+R1*cos(theta2/4)
92 K, 80, R2, R1+wf/2-R1*cos(theta1), 0
93 K, 81, 0, R1+wf/2-R1*cos(theta1), R2
94 K, 82, -R2, R1+wf/2-R1*cos(theta1), 0
95 K, 83, 0, R1+wf/2-R1*cos(theta1), -R2
```

```
K, 84, R2, -R1-wf/2+R1*cos(theta1), 0
K, 85, 0, -R1-wf/2+R1*cos(theta1), R2
K, 86, -R2, -R1-wf/2+R1*cos(theta1), 0
K, 87, 0, -R1-wf/2+R1*cos(theta1), -R2
K, 88, R2*.707, R1+wf/2-R1*cos(theta1), R2*.707
K, 89, -R2*.707, R1+wf/2-R1*cos(theta1), R2*.707
K, 90, R2*.707, R1+wf/2-R1*cos(theta1), -R2*.707
K, 91, -R2*.707, R1+wf/2-R1*cos(theta1), -R2*.707
K, 92, R2*.707, -R1-wf/2+R1*cos(theta1), R2*.707
K, 93, -R2*.707, -R1-wf/2+R1*cos(theta1), R2*.707
K, 94, R2*.707, -R1-wf/2+R1*cos(theta1), -R2*.707
K, 95, -R2*.707, -R1-wf/2+R1*cos(theta1), -R2*.707
K, 96, -xlength+R1*sin(theta2/8), -R1-wf/2+R1*cos(theta2/8), ...
    ↪ -R1-hf/2+R1*cos(theta2/8)
K, 97, -xlength+R1*sin(theta2/8), -R1-wf/2+R1*cos(theta2/8), ...
    ↪ R1+hf/2-R1*cos(theta2/8)
K, 98, -xlength+R1*sin(theta2/8), R1+wf/2-R1*cos(theta2/8), ...
    ↪ R1+hf/2-R1*cos(theta2/8)
K, 99, -xlength+R1*sin(theta2/8), R1+wf/2-R1*cos(theta2/8), ...
    ↪ -R1-hf/2+R1*cos(theta2/8)
K, 100, -xlength+R1*sin(theta2/3), -R1-wf/2+R1*cos(theta2/3), ...
    ↪ -R1-hf/2+R1*cos(theta2/3)
K, 101, -xlength+R1*sin(theta2/3), -R1-wf/2+R1*cos(theta2/3), ...
    ↪ R1+hf/2-R1*cos(theta2/3)
K, 102, -xlength+R1*sin(theta2/3), R1+wf/2-R1*cos(theta2/3), ...
    ↪ R1+hf/2-R1*cos(theta2/3)
K, 103, -xlength+R1*sin(theta2/3), R1+wf/2-R1*cos(theta2/3), ...
    ↪ -R1-hf/2+R1*cos(theta2/3)
K, 104, -xlength+R1*sin(theta2/1.75), ...
    ↪ -R1-wf/2+R1*cos(theta2/1.75), -R1-hf/2+R1*cos(theta2/1.75)
K, 105, -xlength+R1*sin(theta2/1.75), ...
    ↪ -R1-wf/2+R1*cos(theta2/1.75), R1+hf/2-R1*cos(theta2/1.75)
K, 106, -xlength+R1*sin(theta2/1.75), ...
    ↪ R1+wf/2-R1*cos(theta2/1.75), R1+hf/2-R1*cos(theta2/1.75)
K, 107, -xlength+R1*sin(theta2/1.75), ...
    ↪ R1+wf/2-R1*cos(theta2/1.75), -R1-hf/2+R1*cos(theta2/1.75)
K, 108, -xlength+R1*sin(theta2/1.25), ...
    ↪ -R1-wf/2+R1*cos(theta2/1.25), -R1-hf/2+R1*cos(theta2/1.25)
K, 109, -xlength+R1*sin(theta2/1.25), ...
    ↪ -R1-wf/2+R1*cos(theta2/1.25), R1+hf/2-R1*cos(theta2/1.25)
K, 110, -xlength+R1*sin(theta2/1.25), ...
    ↪ R1+wf/2-R1*cos(theta2/1.25), R1+hf/2-R1*cos(theta2/1.25)
K, 111, -xlength+R1*sin(theta2/1.25), ...
    ↪ R1+wf/2-R1*cos(theta2/1.25), -R1-hf/2+R1*cos(theta2/1.25)
K, 112, xlength-R1*sin(theta2/8), -R1-wf/2+R1*cos(theta2/8), ...
    ↪ -R1-hf/2+R1*cos(theta2/8)
K, 113, xlength-R1*sin(theta2/8), -R1-wf/2+R1*cos(theta2/8), ...
    ↪ R1+hf/2-R1*cos(theta2/8)
K, 114, xlength-R1*sin(theta2/8), R1+wf/2-R1*cos(theta2/8), ...
    ↪ R1+hf/2-R1*cos(theta2/8)
K, 115, xlength-R1*sin(theta2/8), R1+wf/2-R1*cos(theta2/8), ...
    ↪ -R1-hf/2+R1*cos(theta2/8)
K, 116, xlength-R1*sin(theta2/3), -R1-wf/2+R1*cos(theta2/3), ...
    ↪ -R1-hf/2+R1*cos(theta2/3)
```

```
K, 117, xlength-R1*sin(theta2/3), -R1-wf/2+R1*cos(theta2/3), ...
    ↪ R1+hf/2-R1*cos(theta2/3)
K, 118, xlength-R1*sin(theta2/3), R1+wf/2-R1*cos(theta2/3), ...
    ↪ R1+hf/2-R1*cos(theta2/3)
K, 119, xlength-R1*sin(theta2/3), R1+wf/2-R1*cos(theta2/3), ...
    ↪ -R1-hf/2+R1*cos(theta2/3)
K, 120, xlength-R1*sin(theta2/1.75), ...
    ↪ -R1-wf/2+R1*cos(theta2/1.75), -R1-hf/2+R1*cos(theta2/1.75)
K, 121, xlength-R1*sin(theta2/1.75), ...
    ↪ -R1-wf/2+R1*cos(theta2/1.75), R1+hf/2-R1*cos(theta2/1.75)
K, 122, xlength-R1*sin(theta2/1.75), R1+wf/2-R1*cos(theta2/1.75), ...
    ↪ R1+hf/2-R1*cos(theta2/1.75)
K, 123, xlength-R1*sin(theta2/1.75), R1+wf/2-R1*cos(theta2/1.75), ...
    ↪ -R1-hf/2+R1*cos(theta2/1.75)
K, 124, xlength-R1*sin(theta2/1.25), ...
    ↪ -R1-wf/2+R1*cos(theta2/1.25), -R1-hf/2+R1*cos(theta2/1.25)
K, 125, xlength-R1*sin(theta2/1.25), ...
    ↪ -R1-wf/2+R1*cos(theta2/1.25), R1+hf/2-R1*cos(theta2/1.25)
K, 126, xlength-R1*sin(theta2/1.25), R1+wf/2-R1*cos(theta2/1.25), ...
    ↪ R1+hf/2-R1*cos(theta2/1.25)
K, 127, xlength-R1*sin(theta2/1.25), R1+wf/2-R1*cos(theta2/1.25), ...
    ↪ -R1-hf/2+R1*cos(theta2/1.25)


!Now to create all of the lines with those keypoints.  Thats a ...
    ↪ bunch of keypoints btw.  Don't mix any of them up...

!Note that I remind myself which line is which by comments,  ...
    ↪ since I can't explicitly label them in the code,  and there ...
    ↪ are too many to just remember.

!Straight Lines: L, Keypoint1, Keypoint2
!Arcs: LARC, P1, P2, Pmid

L, 1, 2          !1
L, 2, 3          !2
L, 3, 4          !3
L, 4, 40             !4
L, 40, 39            !5
L, 39, 38            !6
L, 38, 1             !7
L, 22, 23            !8
L, 21, 41            !9
L, 41, 42            !10
L, 42, 43            !11
L, 43, 24            !12
L, 26, 27            !13
L, 25, 44            !14
L, 44, 45            !15
L, 45, 46            !16
L, 46, 28            !17
L, 5, 6          !18
L, 6, 7          !19
L, 7, 8          !20
```

```
169 L, 9, 10            !21
170 L, 10, 11           !22
171 L, 11, 12           !23
172 L, 12, 9            !24
173 L, 13, 14           !25
174 L, 14, 15           !26
175 L, 15, 16           !27
176 L, 16, 13           !28
177 L, 30, 47           !29
178 L, 47, 48           !30
179 L, 48, 49           !31
180 L, 49, 31           !32
181 L, 29, 32           !33
182 L, 34, 50           !34
183 L, 50, 51           !35
184 L, 51, 52           !36
185 L, 52, 35           !37
186 L, 33, 36           !38
187 L, 18, 53           !39
188 L, 53, 54           !40
189 L, 54, 55           !41
190 L, 55, 19           !42
191 L, 17, 20           !43
192 L, 17, 18           !44
193 L, 19, 20           !45
194 L, 21, 22           !46
195 L, 23, 24           !47
196 L, 25, 26           !48
197 L, 27, 28           !49
198 L, 29, 30           !50
199 L, 31, 32           !51
200 L, 33, 34           !52
201 L, 35, 36           !53
202 LARC, 1, 56, 96     !54
203 LARC, 2, 57, 97     !55
204 LARC, 3, 58, 98     !56
205 LARC, 4, 59, 99     !57
206 LARC, 21, 60, 104       !58
207 LARC, 22, 61, 105       !59
208 LARC, 23, 62, 106       !60
209 LARC, 24, 63, 107       !61
210 LARC, 25, 5, 64     !62
211 LARC, 26, 6, 65     !63
212 LARC, 27, 7, 66     !64
213 LARC, 28, 8, 67     !65
214 LARC, 13, 29, 68        !66
215 LARC, 14, 30, 69        !67
216 LARC, 15, 31, 70        !68
217 LARC, 16, 32, 71        !69
218 LARC, 29, 72, 124       !70
219 LARC, 30, 73, 125       !71
220 LARC, 31, 74, 126       !72
221 LARC, 32, 75, 127       !73
222 LARC, 76, 17, 112       !74
```

```
223 LARC, 77, 18, 113       !75
224 LARC, 78, 19, 114       !76
225 LARC, 79, 20, 115       !77
226 L, 8, 5         !78
227 L, 6, 10            !79
228 L, 7, 11            !80
229 L, 5, 9         !81
230 L, 8, 12            !82
231 L, 10, 14           !83
232 L, 11, 15           !84
233 L, 9, 13            !85
234 L, 12, 16           !86
235 L, 81, 85           !87
236 L, 83, 87           !88
237 LARC, 81, 80, 88        !89
238 LARC, 81, 82, 89        !90
239 LARC, 85, 84, 92        !91
240 LARC, 85, 86, 93        !92
241 L, 81, 11           !93
242 L, 83, 12           !94
243 L, 85, 10           !95
244 L, 87, 9            !96
245 LARC, 80, 83, 90        !97
246 LARC, 82, 83, 91        !98
247 LARC, 84, 87, 94        !99
248 LARC, 86, 87, 95        !100
249 L, 1, 4         !101
250 L, 21, 24           !102
251 L, 25, 28           !103
252 L, 30, 31           !104
253 L, 34, 35           !105
254 L, 18, 19           !106
255 LCOMB, 89, 97, 1        !107
256 LCOMB, 90, 98, 1        !108
257 LCOMB, 91, 99, 1        !109
258 LCOMB, 92, 100, 1       !110
259 LARC, 56, 21, 100       !111
260 LARC, 57, 22, 101       !112
261 LARC, 58, 23, 102       !113
262 LARC, 59, 24, 103       !114
263 LARC, 60, 25, 108       !115
264 LARC, 61, 26, 109       !116
265 LARC, 62, 27, 110       !117
266 LARC, 63, 28, 111       !118
267 LARC, 76, 33, 116       !119
268 LARC, 77, 34, 117       !120
269 LARC, 78, 35, 118       !121
270 LARC, 79, 36, 119       !122
271 LARC, 72, 33, 120       !123
272 LARC, 73, 34, 121       !124
273 LARC, 74, 35, 122       !125
274 LARC, 75, 36, 123       !126
275 L, 56, 57           !127
276 L, 57, 58           !128
```

```
277 L, 58, 59           !129
278 L, 59, 56           !130
279 L, 60, 61           !131
280 L, 61, 62           !132
281 L, 62, 63           !133
282 L, 63, 60           !134
283 L, 76, 77           !135
284 L, 77, 78           !136
285 L, 78, 79           !137
286 L, 79, 76           !138
287 L, 72, 73           !139
288 L, 73, 74           !140
289 L, 74, 75           !141
290 L, 75, 72           !142
291
292
293
294 !Now that lines have been defined,  create the areas.  Don't mix ...
       ↪ up the lines or keypoints now...
295
296 !AL, L1, L2, L3, L4, L5, L6
297
298 !Number the areas the same way as the lines
299 !Flat areas
300 AL, 1, 2, 3, 101            !1
301 AL, 46, 8, 47, 102          !2
302 AL, 48, 13, 49, 103         !3
303 AL, 18, 19, 20, 78          !4
304 AL, 22, 95, 87, 93          !5
305 AL, 25, 26, 27, 28          !6
306 AL, 33, 51, 104, 50         !7
307 AL, 105, 53, 38, 52         !8
308 AL, 106, 45, 43, 44         !9
309 AL, 78, 81, 24, 82          !10
310 AL, 24, 86, 28, 85          !11
311 AL, 19, 80, 22, 79          !12
312 AL, 22, 84, 26, 83          !13
313 AL, 79, 95, 110, 96, 81, 18     !14
314 AL, 80, 93, 82, 94, 108, 20     !15
315 AL, 83, 25, 85, 95, 109, 96     !16
316 AL, 84, 27, 107, 93, 86, 94     !17
317
318
319 !Now the Curved Areas
320
321 AL, 54, 127, 55, 1              !18
322 AL, 55, 128, 56, 2              !19
323 AL, 56, 129, 57, 3              !20
324 AL, 57, 130, 54, 101            !21
325 AL, 58, 131, 59, 46             !22
326 AL, 59, 132, 60, 8              !23
327 AL, 60, 133, 61, 47             !24
328 AL, 61, 134, 58, 102            !25
329 AL, 62, 18, 63, 48              !26
```

```
AL, 63, 19, 64, 13                !27
AL, 64, 20, 65, 49                !28
AL, 65, 78, 62, 103               !29
AL, 66, 50, 67, 25                !30
AL, 67, 104, 68, 26               !31
AL, 68, 51, 69, 27                !32
AL, 69, 33, 66, 28                !33
AL, 70, 139, 71, 50               !34
AL, 71, 140, 72, 104              !35
AL, 72, 141, 73, 51               !36
AL, 73, 142, 70, 33               !37
AL, 74, 135, 75, 44               !38
AL, 75, 136, 76, 106              !39
AL, 76, 137, 77, 45               !40
AL, 77, 138, 74, 43               !41
AL, 24, 96, 88, 94                !42
AL, 87, 109, 88, 107              !43
AL, 87, 110, 88, 108              !44
AL, 127, 128, 129, 130            !45
AL, 131, 132, 133, 134            !46
AL, 135, 136, 137, 138            !47
AL, 139, 140, 141, 142            !48

!Flat and Curved Areas section 2
AL, 127, 111, 112, 46             !49
AL, 128, 112, 113, 8              !50
AL, 129, 113, 114, 47             !51
AL, 130, 114, 111, 102            !52
AL, 131, 48, 115, 116             !53
AL, 132, 13, 116, 117             !54
AL, 133, 49, 117, 118             !55
AL, 134, 103, 118, 115            !56
AL, 52, 139, 123, 124             !57
AL, 105, 140, 124, 125            !58
AL, 53, 141, 125, 126             !59
AL, 38, 142, 126, 123             !60
AL, 52, 135, 119, 120             !61
AL, 105, 136, 120, 121            !62
AL, 53, 137, 121, 122             !63
AL, 38, 138, 122, 119             !64



!Now for the volumes,  don't mix any of the lines or keypoints or ...
    ↪ areas up...

!VA, A1, A2, A3, A4..

VA, 1, 45, 18, 19, 20, 21           !1
VA, 2, 46, 22, 23, 24, 25           !2
VA, 3, 4, 26, 27, 28, 29            !3
VA, 4, 5, 42, 44, 10, 12, 14, 15        !4
VA, 5, 42, 43, 6, 11, 13, 16, 17        !5
VA, 6, 7, 30, 31, 32, 33            !6
```

```
VA, 7, 48, 34, 35, 36, 37           !7
VA, 47, 9, 38, 39, 40, 41           !8
VA, 49, 50, 51, 52, 45, 2           !9
VA, 53, 54, 55, 56, 46, 3           !10
VA, 57, 58, 59, 60, 48, 8           !11
VA, 61, 62, 63, 64, 47, 8           !12

!LSEL, S, LINE, , 4, 7
!LDELE, ALL
!LSEL, S, LINE, , 39, 42
!LDELE, ALL

!KSEL, S, KP, , 38, 40,
!KDELE, ALL
!KSEL, S, KP, , 53, 55
!KDELE, ALL
!Add the Volumes up to make the total volume
!set ans_consec=yes
BOPTN, KEEP, YES
VADD, ALL
SHPP, OFF, ALL
!Error is right here,  with VMESH,  need to say yes to the error ...
    ↪ or just not check it and do it.
VMESH, 13
NSEL, ALL
NWRITE, E:\Research\Trials\Combined\TRIALS4, txt, , 0
EWRITE, E:\Research\Trials\Combined\TRIALS5, txt, , 0
FINISH
/SOLU
!Add forces to the requisite keypoints with the FK command
!FK, Keypoint, ForceDirection, ForceApplied

!Loading condition 1,  and solution 1 (Will be 3 solutions)

FK, 4, FZ, -Force/2
FK, 1, FZ, -Force/2

FK, 18, FZ, Force/2
FK, 19, FZ, Force/2

!Keep area constant with DA command
DA, 43, UX, 0
DA, 43, UY, 0
DA, 43, UZ, 0
DA, 44, UX, 0
DA, 44, UY, 0
DA, 44, UZ, 0

allsel, all, all

SOLVE, , , , , NOCHECK
FINISH
/POST1
PLVECT, U, , , , VECT, NODE
```

```
436 /OUTPUT, E:\Research\Trials\Combined\TRIALS3, txt, ,
437 ETABLE, U1X, U, X
438 ETABLE, U1Z, U, Z
439 !Other 2 ouputs use /OUTPUT, ...
       ↪ E:\Research\Trials\Combined\TRIALS32, txt, ,  and TRIALS31
440 PRETAB
441 /OUT
442 FINISH
443
444
445
446 !Case 2--------------------------------------
447
448 /SOLU
449 !Add forces to the requisite keypoints with the FK command
450 !FK, Keypoint, ForceDirection, ForceApplied
451
452 !Loading condition 1,  and solution 1 (Will be 3 solutions)
453
454 FKDELE, 4, ALL
455 FKDELE, 1, ALL
456 FKDELE, 18, ALL
457 FKDELE, 19, ALL
458
459
460 FK, 56, FZ, -Force/2*cos(theta2/4)
461 FK, 56, FX, Force/2*sin(theta2/4)
462 FK, 59, FZ, -Force/2*cos(theta2/4)
463 FK, 59, FX, Force/2*sin(theta2/4)
464
465 FK, 77, FZ, Force/2*cos(theta2/4)
466 FK, 77, FX, -Force/2*sin(theta2/4)
467 FK, 78, FZ, Force/2*cos(theta2/4)
468 FK, 78, FX, -Force/2*sin(theta2/4)
469
470 allsel, all, all
471
472 SOLVE, , , , , NOCHECK
473 FINISH
474 /POST1
475 PLVECT, U, , , , VECT, NODE
476 /OUTPUT, E:\Research\Trials\Combined\TRIALS6, txt, ,
477 ETABLE, U1X, U, X
478 ETABLE, U1Z, U, Z
479 !Other 2 ouputs use /OUTPUT, ...
       ↪ E:\Research\Trials\Combined\TRIALS32, txt, ,  and TRIALS31
480 PRETAB
481 /OUT
482 FINISH
483
484
485
486 !Case 3-----------------------------------
487
```

```
488 /SOLU
489 !Add forces to the requisite keypoints with the FK command
490 !FK, Keypoint, ForceDirection, ForceApplied
491
492 !Loading condition 1,  and solution 1 (Will be 3 solutions)
493
494 FKDELE, 56, ALL
495 FKDELE, 59, ALL
496 FKDELE, 77, ALL
497 FKDELE, 78, ALL
498
499
500 FK, 21, FZ, -Force/2*cos(theta2/2)
501 FK, 21, FX, Force/2*sin(theta2/2)
502 FK, 24, FZ, -Force/2*cos(theta2/2)
503 FK, 24, FX, Force/2*sin(theta2/2)
504
505 FK, 34, FZ, Force/2*cos(theta2/2)
506 FK, 34, FX, -Force/2*sin(theta2/2)
507 FK, 35, FZ, Force/2*cos(theta2/2)
508 FK, 35, FX, -Force/2*sin(theta2/2)
509
510 allsel, all, all
511
512 SOLVE, , , , , NOCHECK
513 FINISH
514 /POST1
515 PLVECT, U, , , , VECT, NODE
516 /OUTPUT, E:\Research\Trials\Combined\TRIALS7, txt, ,
517 ETABLE, U1X, U, X
518 ETABLE, U1Z, U, Z
519 !Other 2 ouputs use /OUTPUT, ...
       ↪ E:\Research\Trials\Combined\TRIALS32, txt, ,  and TRIALS31
520 PRETAB
521 /OUT
522 FINISH
523
524
525
526 !Case 4-------------------------------------
527
528 /SOLU
529 !Add forces to the requisite keypoints with the FK command
530 !FK, Keypoint, ForceDirection, ForceApplied
531
532 !Loading condition 1,  and solution 1 (Will be 3 solutions)
533
534 FKDELE, 21, ALL
535 FKDELE, 24, ALL
536 FKDELE, 34, ALL
537 FKDELE, 35, ALL
538
539
540 FK, 60, FZ, -Force/2*cos(theta2/1.5)
```

```
FK, 60, FX, Force/2*sin(theta2/1.5)
FK, 63, FZ, -Force/2*cos(theta2/1.5)
FK, 63, FX, Force/2*sin(theta2/1.5)

FK, 73, FZ, Force/2*cos(theta2/1.5)
FK, 73, FX, -Force/2*sin(theta2/1.5)
FK, 74, FZ, Force/2*cos(theta2/1.5)
FK, 74, FX, -Force/2*sin(theta2/1.5)

allsel, all, all

SOLVE, , , , , NOCHECK
FINISH
/POST1
PLVECT, U, , , , VECT, NODE
/OUTPUT, E:\Research\Trials\Combined\TRIALS8, txt, ,
ETABLE, U1X, U, X
ETABLE, U1Z, U, Z
!Other 2 ouputs use /OUTPUT, ...
   ↪ E:\Research\Trials\Combined\TRIALS32, txt, ,  and TRIALS31
PRETAB
/OUT
FINISH



!Case 5---------------------------------

/SOLU
!Add forces to the requisite keypoints with the FK command
!FK, Keypoint, ForceDirection, ForceApplied

!Loading condition 1,  and solution 1 (Will be 3 solutions)

FKDELE, 60, ALL
FKDELE, 63, ALL
FKDELE, 73, ALL
FKDELE, 74, ALL


FK, 25, FZ, -Force/2*cos(theta2)
FK, 25, FX, Force/2*sin(theta2)
FK, 28, FZ, -Force/2*cos(theta2)
FK, 28, FX, Force/2*sin(theta2)

FK, 30, FZ, Force/2*cos(theta2)
FK, 30, FX, -Force/2*sin(theta2)
FK, 31, FZ, Force/2*cos(theta2)
FK, 31, FX, -Force/2*sin(theta2)

allsel, all, all

SOLVE, , , , , NOCHECK
FINISH
```

```
/POST1
PLVECT, U, , , , VECT, NODE
/OUTPUT, E:\Research\Trials\Combined\TRIALS9, txt, ,
ETABLE, U1X, U, X
ETABLE, U1Z, U, Z
!Other 2 ouputs use /OUTPUT, ...
    ↪ E:\Research\Trials\Combined\TRIALS32, txt, ,  and TRIALS31
PRETAB
/OUT
FINISH

/EXIT, NOSAVE,
```

A.4 Code File Names

Below are the file names for the code with a brief description

- Tester11.m = 5 Location MatLab Optimization Function
- Trials11.txt = 5 Location ANSYS Code
- combinedtrial9.m = 9 Location MatLab Optimization Function
- UpdatedAnsys3.txt = 9 Location ANSYS Code
- combinedtrial21.m = 21 Location MatLab Optimization Function
- UpdatedAnsys2.txt = 21 Location MatLab Code
- confun.m = Constraint Function
- runme.m = Optimization Run File - Note the function called needs to be changed to the correct file for the number of locations being optimized

A.5 Free Joint Angle

The flexure has whiskers on the ends of the flexure that extend the flexure past the point where all four contactors can simultaneously be in contact with the flexure. See Figure A.4 to show the whisker and free joint angle.

These whiskers allow the VSA to recapture the flexure after a free joint is provided. To define the free joint angle geometrically an additional variable must be set, the radius of the contact selection system, r_1. This is not an optimized variable, and is instead selected in the VSA design process. The following equations fully describe the free joint angle given the geometric information.

$$A = X + R_1 * \sin(\Theta_d) \tag{A.1}$$

A is the horizontal length of the end location of the connecting bar, shown on Figure 2.7.

$$F = \frac{h_f}{2} + R_1 - R_1 * \cos(\Theta_d) - r_1 \tag{A.2}$$

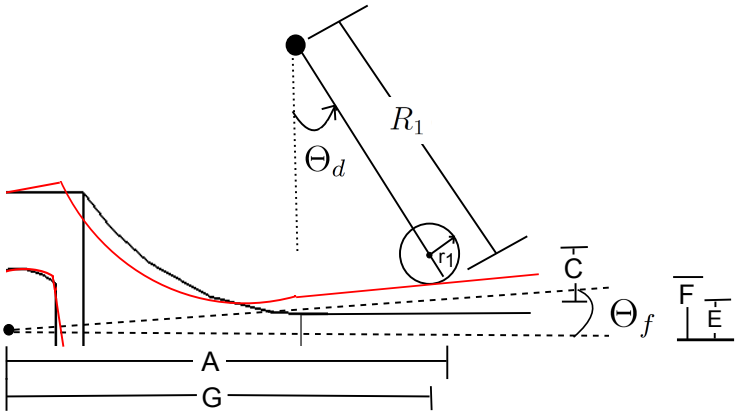


Figure A.4: Free Joint Angle (Θ_f) with Flexure at Initial Position and Maximum Free Joint Position

F is the vertical length component of the connecting bar to the initial centerline.

$$C = \frac{h_f}{2} + r_1 \tag{A.3}$$

C is the minimum distance from the center of the roller to the center line of the flexure when in contact.

$$G = \frac{(F * C + \sqrt{A^4 + A^2 * F^2 - A^2 * C^2})}{A^2 + F^2} \tag{A.4}$$

$$E = \frac{\frac{F * \sqrt{-A^2 * (-A^2 - F^2 + C^2)}}{(A^2 + F^2)} + (\frac{(F^2 * C)}{(A^2 + F^2)} - C)}{A} \tag{A.5}$$

G and E take the resulting triangles, and apply the law of sines and cosines to the triangles formed from the above line segments to get the horizontal and vertical components of the triangle formed with the free joint angle, Θ_f.

$$\Theta_f = \arctan{(\frac{E}{G})} \tag{A.6}$$

Figure A.5 shows different force-deflection curves that the joint will have at various contactor angles. Note that VSA will have a continuous range for free angles and stiffnesses, not just the discrete ranges being shown.

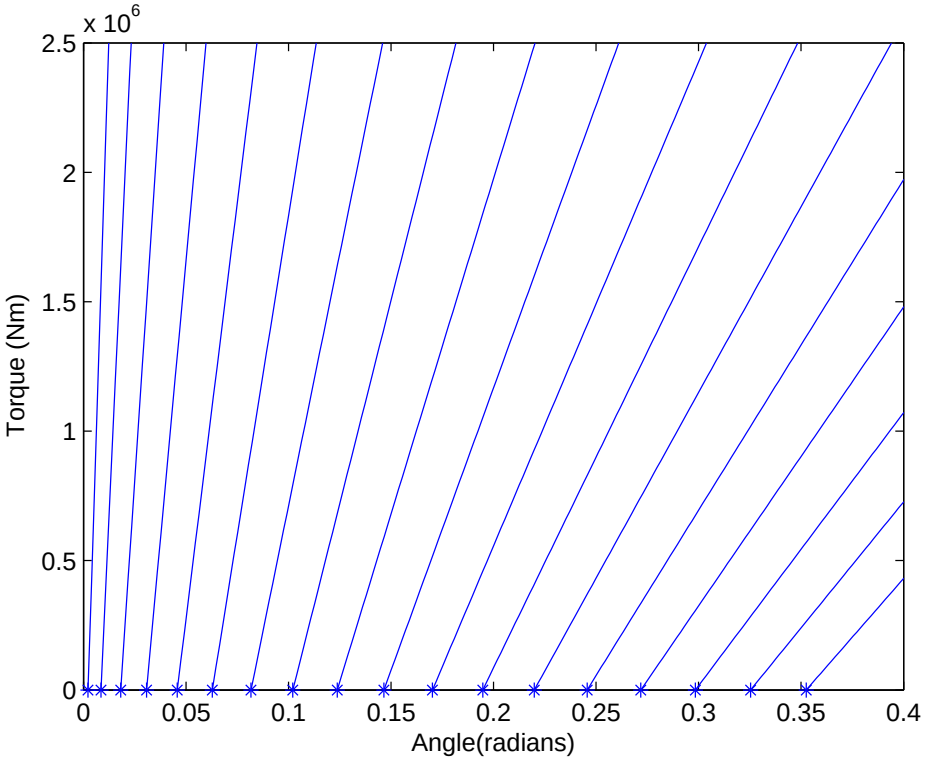


Figure A.5: Force-Angular Joint Deflection Curve for Free Joint Stiffness Selection Angles

APPENDIX B

APPENDIX B: MANUFACTURED PART DRAFTINGS

B.1 Assembly Plan

See below for the assembly plan for the VSA. Refer to the CAD model for orientation and location of all parts.

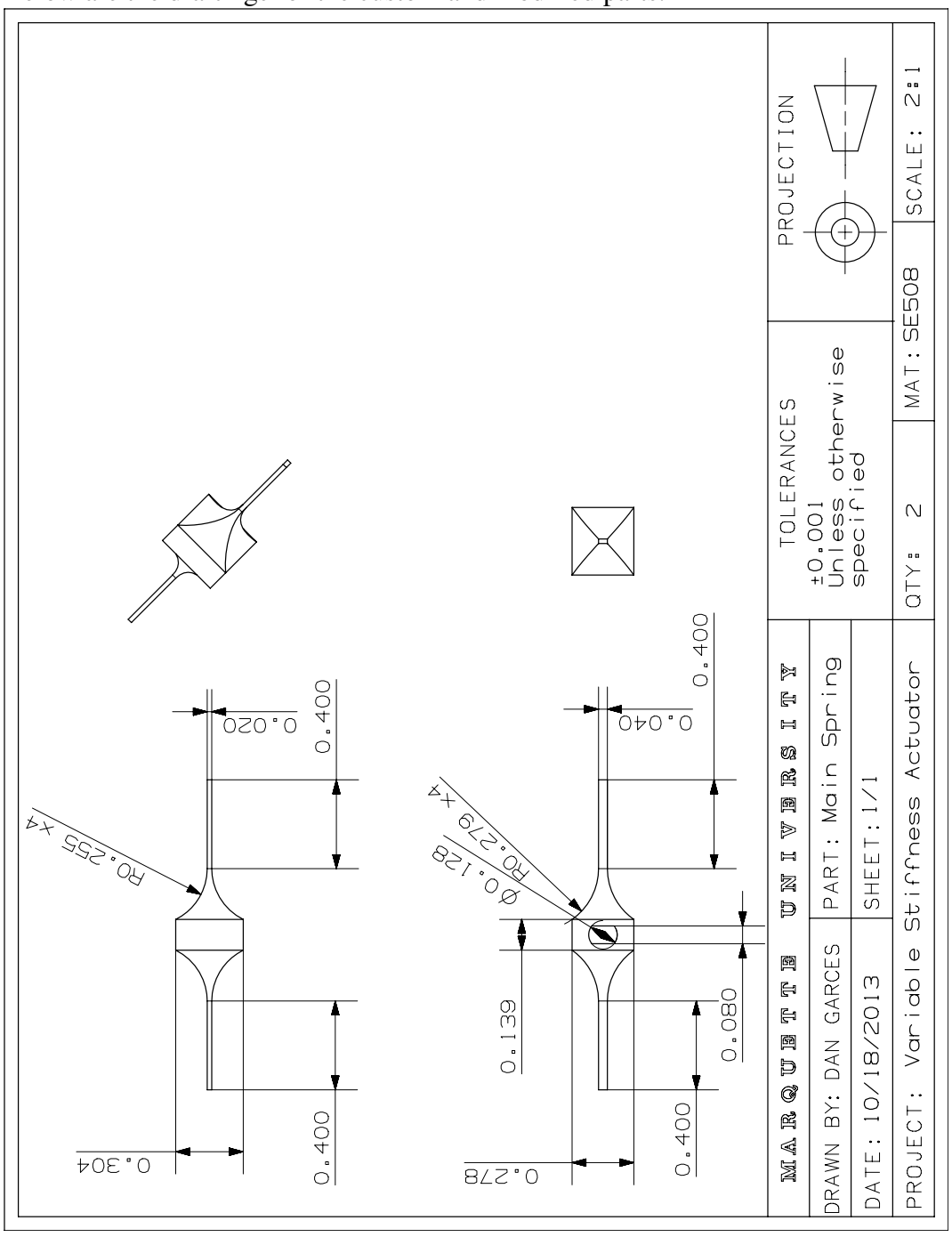
1. Ensure all part dimensions are correct, and all burrs have been removed.
2. Insert Sleeve A into Shaft C and Shaft D
3. Insert Shaft F into Shaft C, D and Sleeve A
4. Add snap ring onto Shaft B
5. Insert Bearing into Plate, Bottom
6. Insert Bearing into Plate, Bottom
7. Insert Bearing into Plate, Middle
8. Insert Bearing into Plate, Top
9. Insert Shaft C and Shaft D into bearings on Plate, Middle
10. Secure Shaft C and Shaft D in place with snap rings
11. Insert Gear N32 onto Shaft D
12. Insert Gear N36 onto Shaft C
13. Add snap ring onto Shaft C and Shaft D
14. Insert Shaft B into bearings on Plate, Top
15. Insert Gear N21 onto Shaft E
16. Add snap rings to Shaft E
17. Insert Shaft E into bearing on Plate, Top
18. Add snap rings to Shaft E
19. Add Bevel Gear (Modified) to Shaft B
20. Secure with Bevel Gear (Modified) with Collar
21. Insert Gear N16 to Shaft B

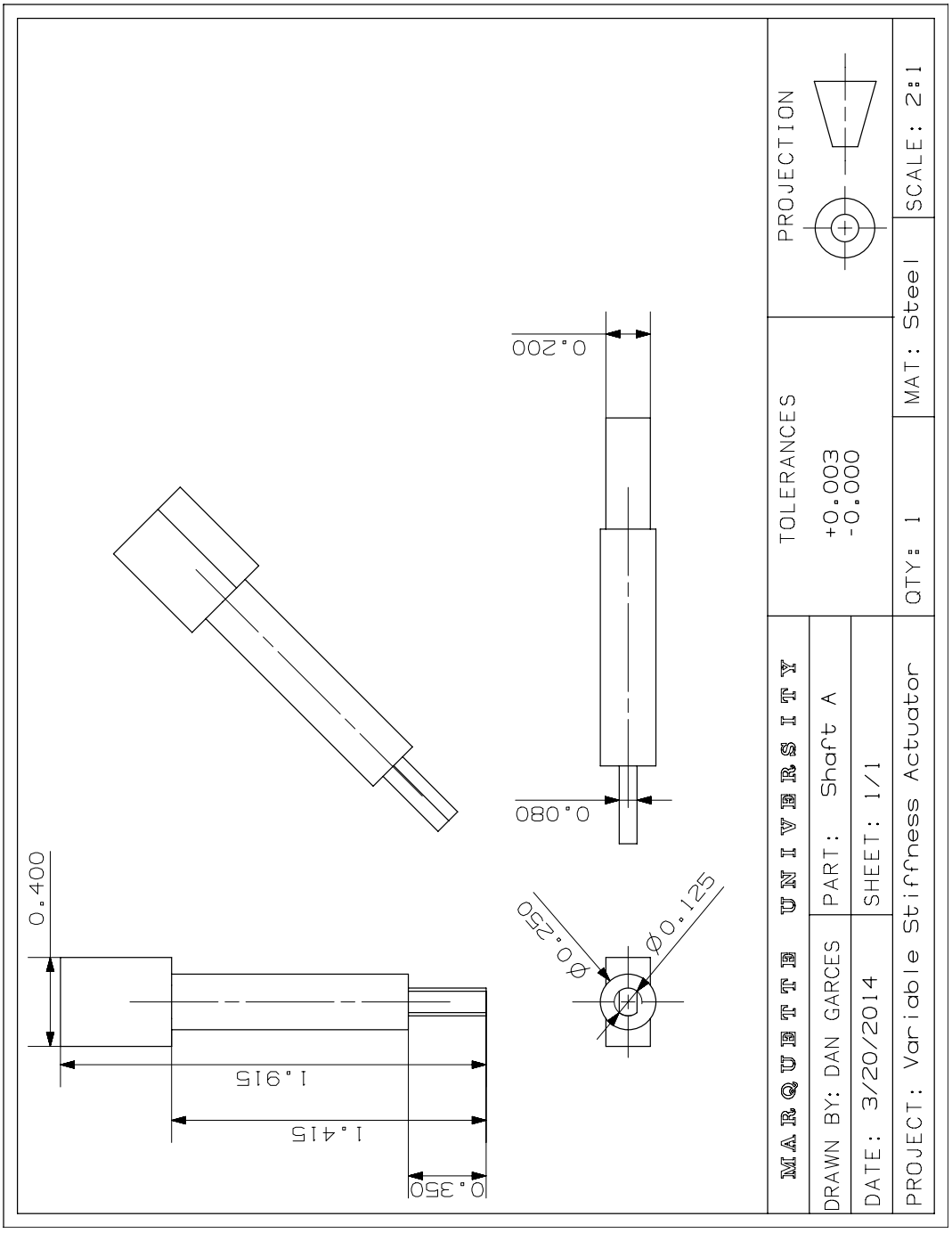
22. Add additional snap rings onto Shaft B
23. Insert Shaft A into bearings on Plate, Bottom
24. Add snap rings to Shaft A
25. Insert Flexure onto Shaft A
26. Add snap rings to Shaft A
27. Insert Plate, Middle into Cylinder, Link i+1
28. Align Plate, Middle to the proper location
29. Add fixturing shafts to Cylinder, Link i+1
30. Attach Plate, Bottom to Cylinder, Link i+1 with screws
31. Insert Gear N18 above Bearing on Plate, Middle, meshing with Gears N36
32. Lower Plate, Top into Cylinder, Link i+1, aligning Shaft B to Gear N18
33. Secure Plate, Top to Cylinder, Link i+1 with screws
34. Remove fixturing shafts
35. Add snap rings to Shaft D at Plate, Top
36. Add snap rings to Shaft C and Shaft D at Plate, Bottom
37. Add Connector Motor Small to Bevel Gear
38. Add Connector Motor Large to Bevel Gear
39. Take Bevel Gear with Connector Motor Small and align with Maxon DCX10L Motor Mounting Location
40. Add epoxy to threads on Maxon DCX10L Motor Mounting Location, wiping off excess
41. Insert Maxon DCX10L Motor into mounting location, ensuring shaft is properly inserted into Connector Motor Small
42. Screw Maxon DCX10L Motor into Cylinder, Link i+1
43. Insert bearings into Cylinder, Link i A
44. Screw Cylinder, Link i A and Cylinder, Link i B together
45. Insert Plate, Bottom into Bearing on Cylinder, Link i A
46. Add snap ring to Plate, Bottom

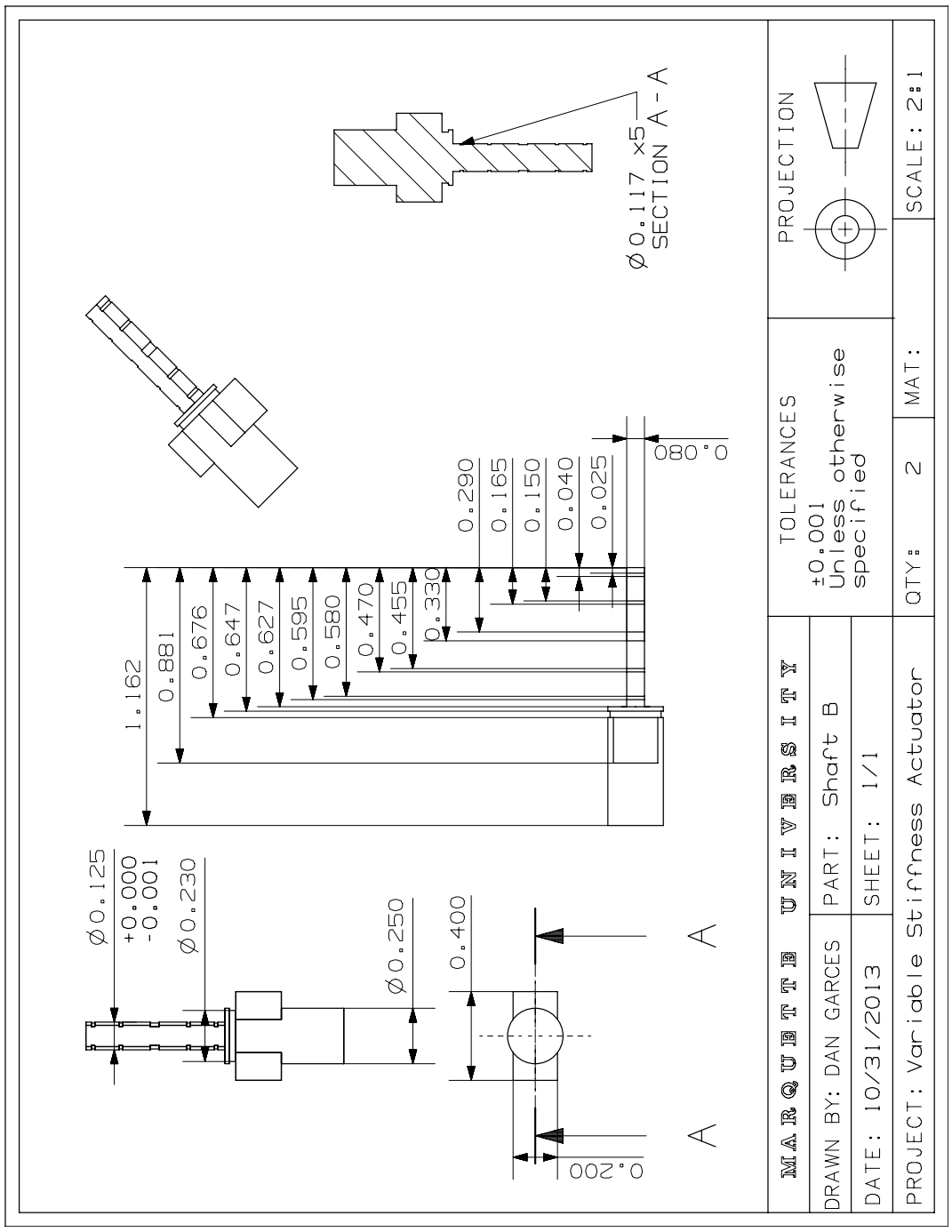
47. Insert bearing into Cylinder, Link i B
48. Insert Connector Motor Large into Bevel Gear
49. Insert Bevel Gear without Connector Motor Large into bearing on Cylinder, Link i B
50. Align Bevel Gear containing Connector Motor Large with Maxon DCX22S Motor Mounting Location
51. Insert Maxon DCX22S Motor into mounting location, ensuring shaft is properly inserted into Connector Motor Large
52. Secure Maxon DCX22S Motor to Cylinder, Link i B with screws

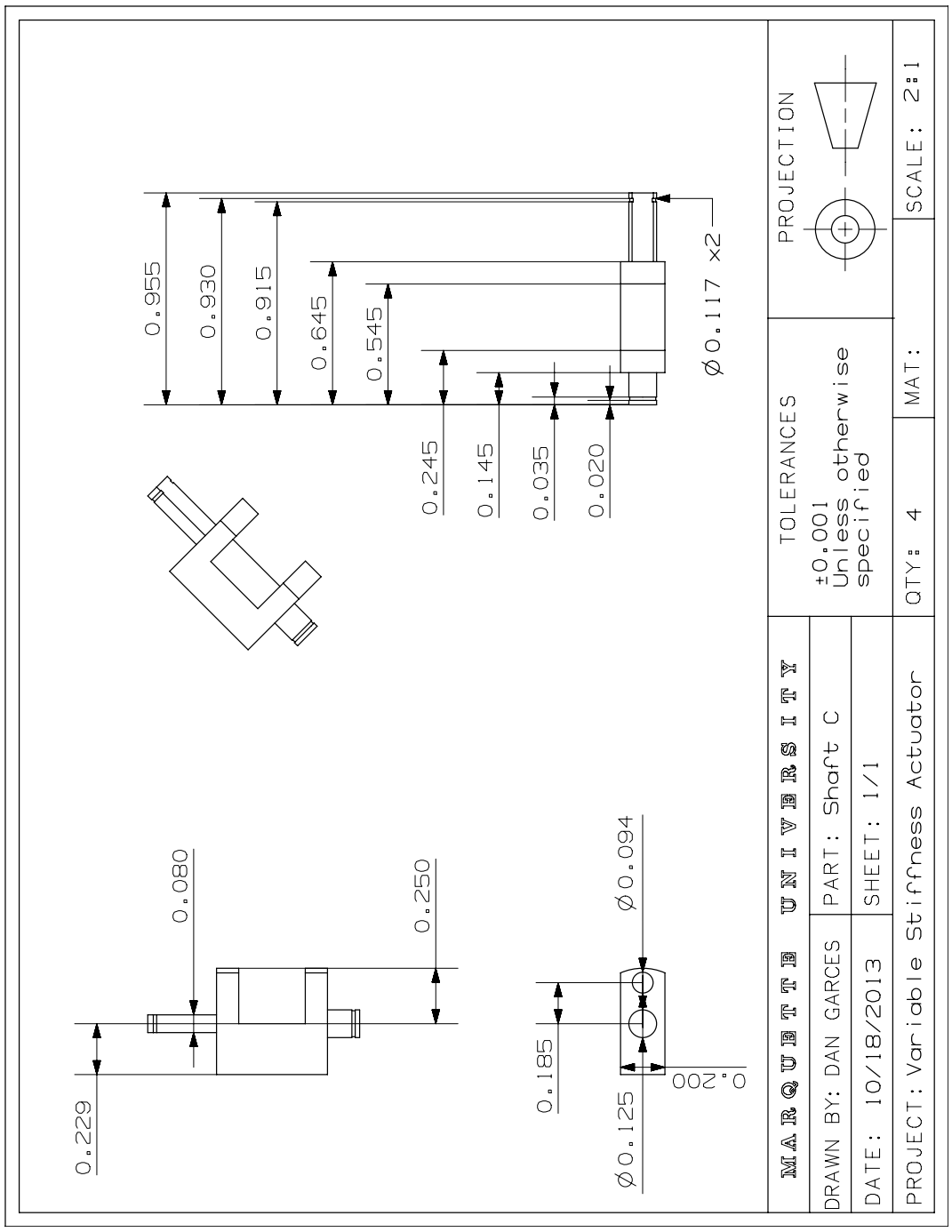
B.2 Draftings

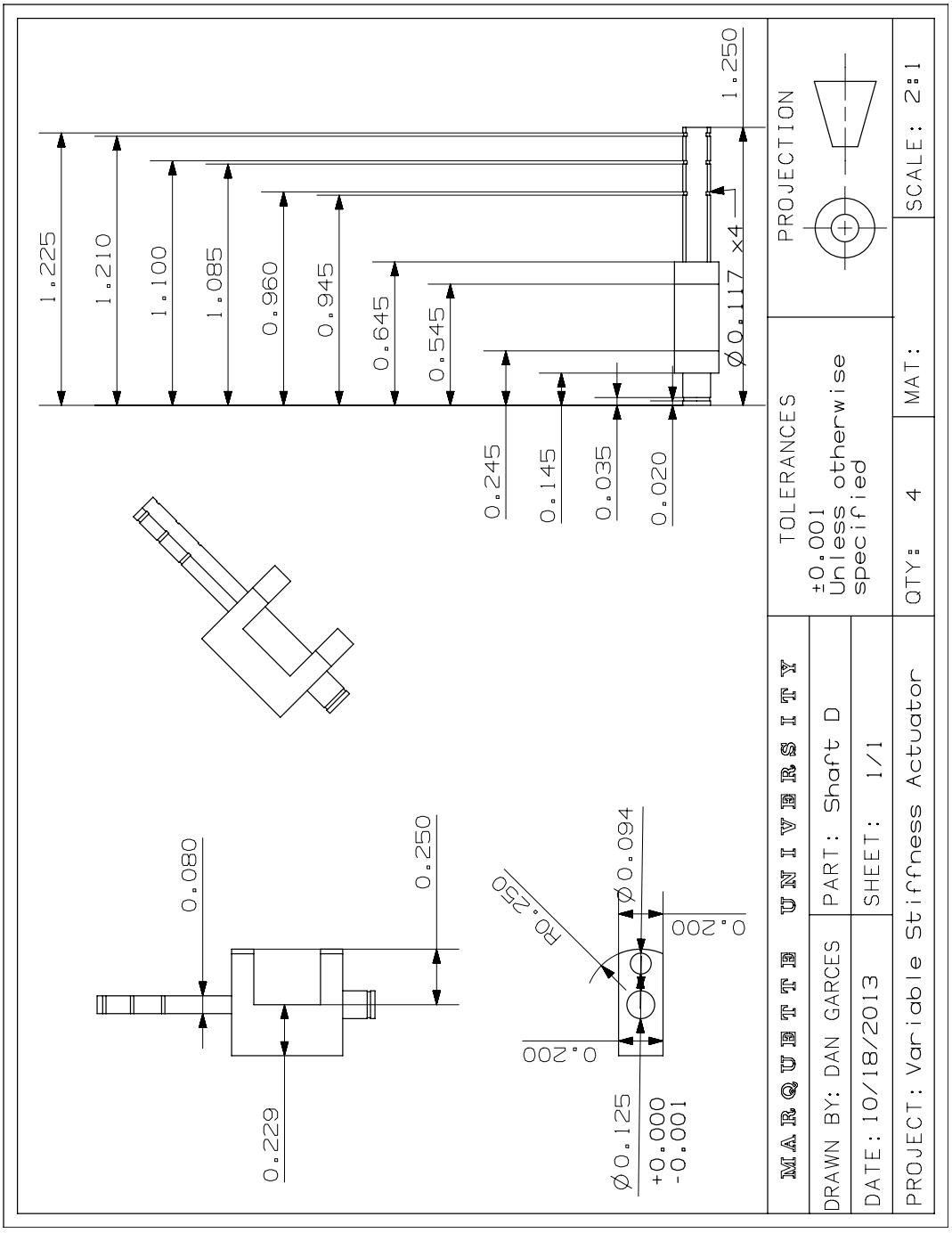
Below are the draftings for the custom and modified parts.

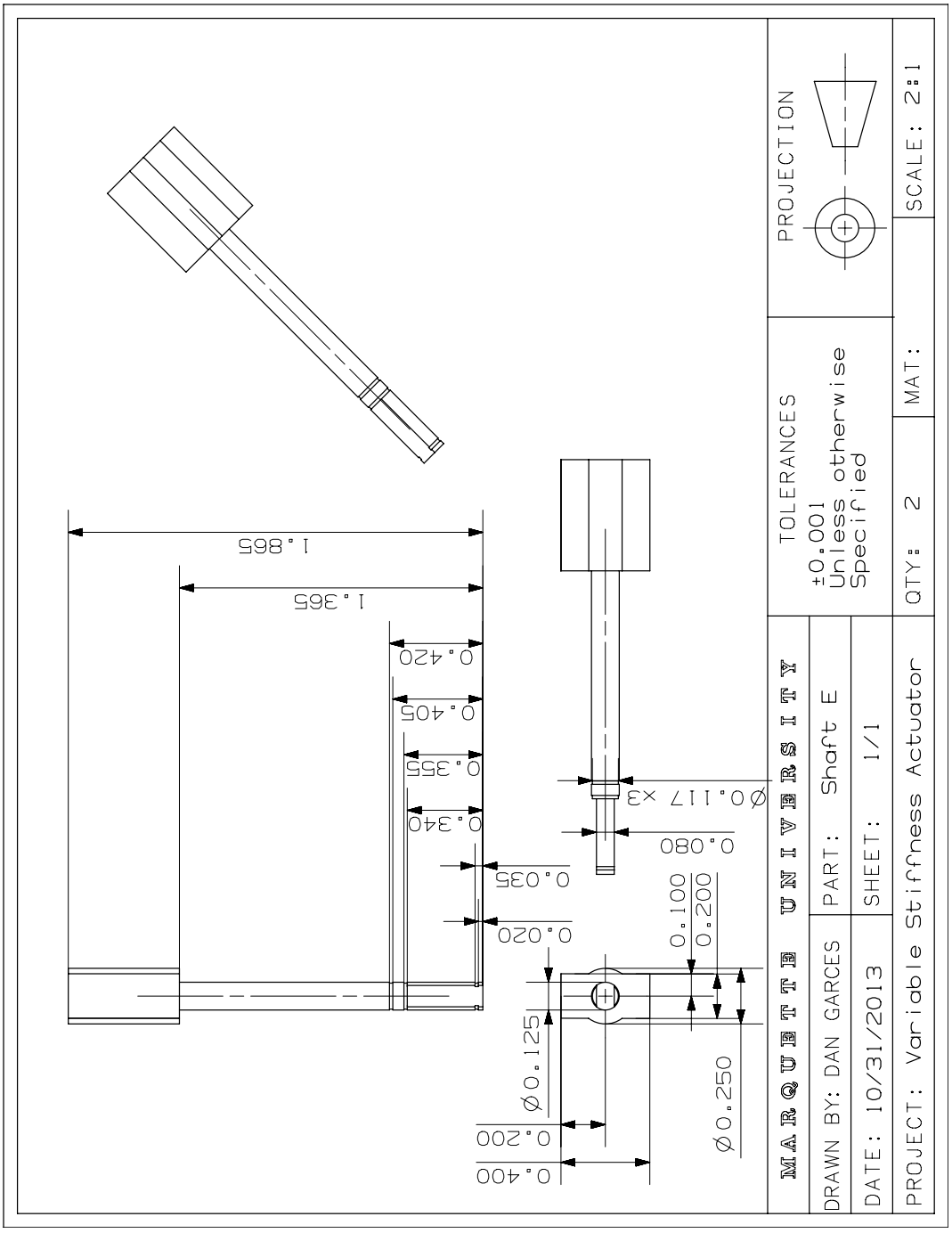


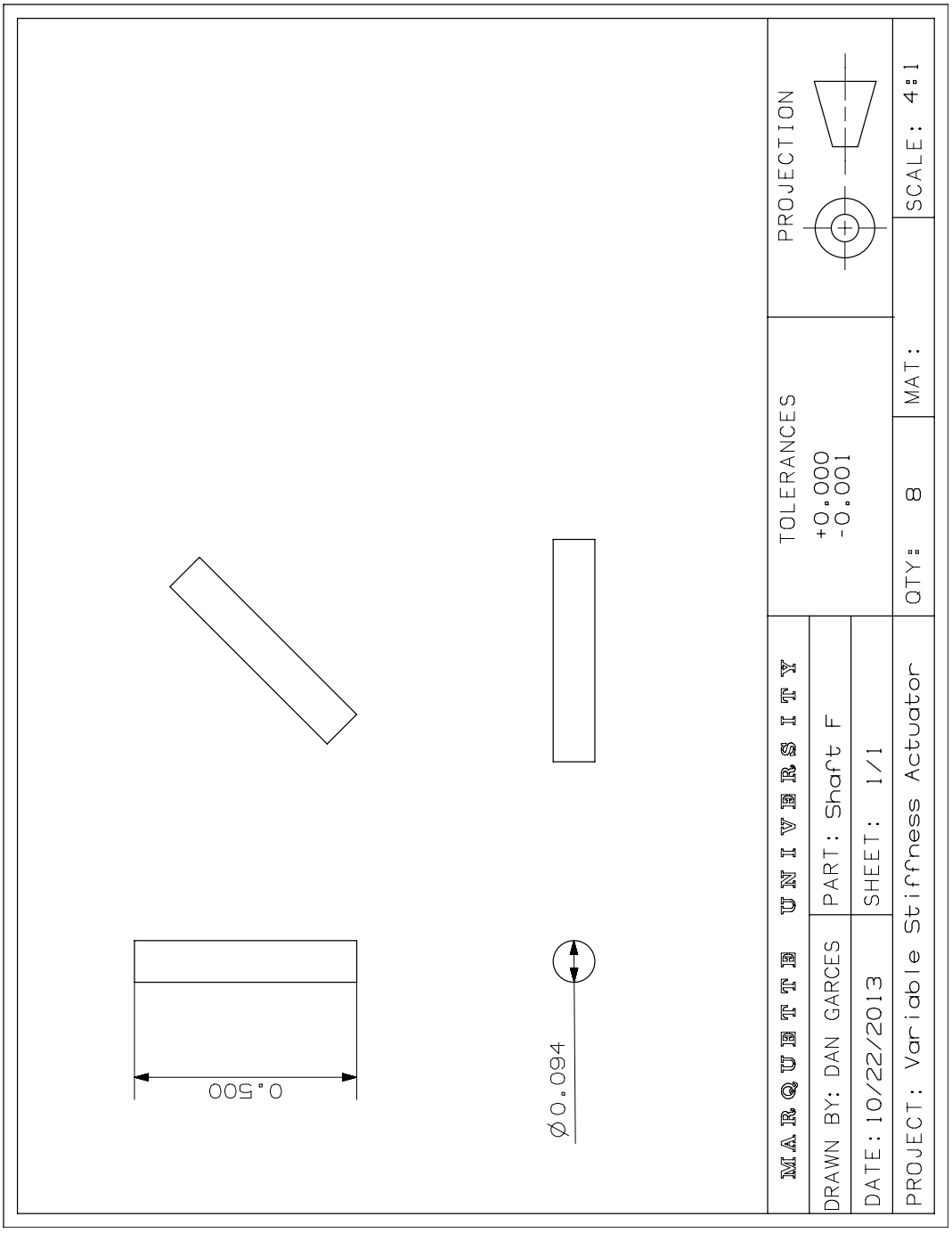

0.400
1.915
1.415
0.350
0.200
0.080
Ø0.250
Ø0.125
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: Shaft A
DATE: 3/20/2014
SHEET: 1/1
PROJECT: Variable Stiffness Actuator
TOLERANCES
+0.003
-0.000
QTY: 1
MAT: Steel
PROJECTION
SCALE: 2:1

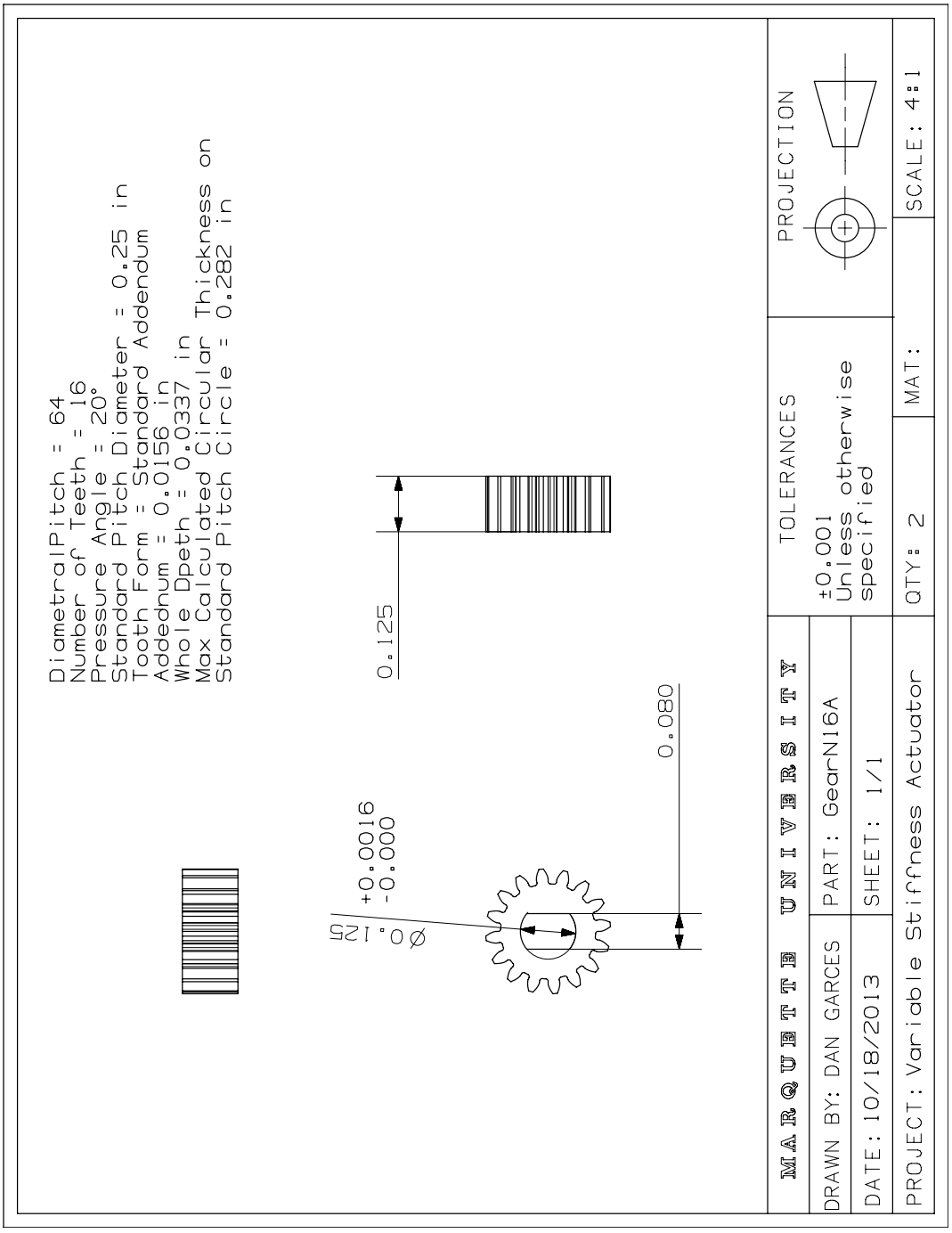

Ø0.117 ×5
SECTION A - A
1.162
0.881
0.676
0.647
0.627
0.595
0.580
0.470
0.455
0.330
0.290
0.165
0.150
0.040
0.025
0.000
0.125
+0.000
-0.001
Ø0.230
Ø0.250
0.400
0.200
A
A
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: Shaft B
DATE: 10/31/2013
SHEET: 1/1
PROJECT: Variable Stiffness Actuator
TOLERANCES
±0.001
Unless otherwise specified
QTY: 2
MAT:
PROJECTION
SCALE: 2:1


0.955
0.930
0.915
0.645
0.545
0.245
0.145
0.035
0.020
Ø0.117 ×2
0.080
0.250
0.229
Ø0.094
0.185
Ø0.125
0.200
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: Shaft C
DATE: 10/18/2013
SHEET: 1/1
PROJECT: Variable Stiffness Actuator
TOLERANCES
±0.001
Unless otherwise specified
PROJECTION
QTY: 4
MAT:
SCALE: 2:1

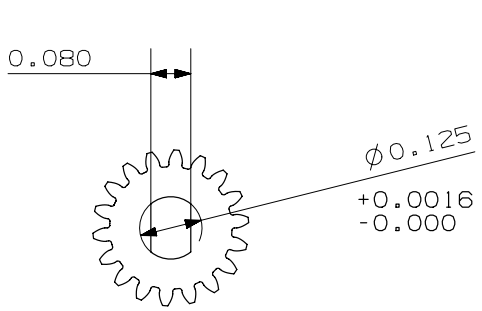

1.225
1.210
1.100
1.085
0.960
0.945
0.645
0.545
0.245
0.145
0.035
0.020
Ø0.117 x4
1.250
0.080
0.250
0.229
R0.250
Ø0.094
0.000
Ø0.125 +0.000 -0.001
TOLERANCES
±0.001
Unless otherwise specified
PROJECTION
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: Shaft D
DATE: 10/18/2013
SHEET: 1/1
QTY: 4
MAT:
SCALE: 2:1
PROJECT: Variable Stiffness Actuator

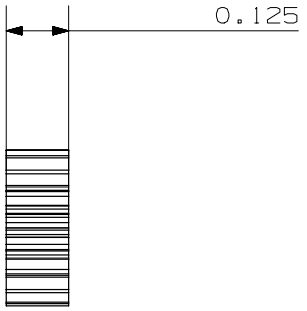

MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
DATE: 10/31/2013
PART: Shaft E
SHEET: 1/1
PROJECT: Variable Stiffness Actuator
TOLERANCES
±0.001
Unless otherwise
Specified
QTY: 2
MAT:
PROJECTION
SCALE: 2:1
1.865
1.365
0.420
0.405
0.355
0.340
0.035
0.020
0.080
Ø0.117 x3
0.100
0.200
Ø0.125
0.200
0.400
Ø0.250


0.500
Ø0.094
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: Shaft F
DATE: 10/22/2013
SHEET: 1/1
PROJECT: Variable Stiffness Actuator
TOLERANCES
+0.000
-0.001
QTY: 8
MAT:
PROJECTION
SCALE: 4:1

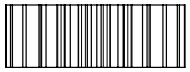

DiametralPitch = 64
Number of Teeth = 16
Pressure Angle = 20°
Standard Pitch Diameter = 0.25 in
Tooth Form = Standard Addendum
Addednum = 0.0156 in
Whole Dpeth = 0.0337 in
Max Calculated Circular Thickness on
Standard Pitch Circle = 0.282 in
0.125
0.080
Ø.1250 +0.0016 -0.0000
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: GearN16A
DATE: 10/18/2013
SHEET: 1/1
PROJECT: Variable Stiffness Actuator
TOLERANCES
±0.001
Unless otherwise specified
QTY: 2
MAT:
PROJECTION
SCALE: 4:1

DiametralPitch = 64
Number of Teeth = 18
Pressure Angle = 20°
Standard Pitch Diameter = 0.282 in
Tooth Form = Standard Addendum
Addednum = 0.0156 in
Whole Dpeth = 0.0337 in
Max Calculated Circular Thickness on
Standard Pitch Circle = 0.313 in

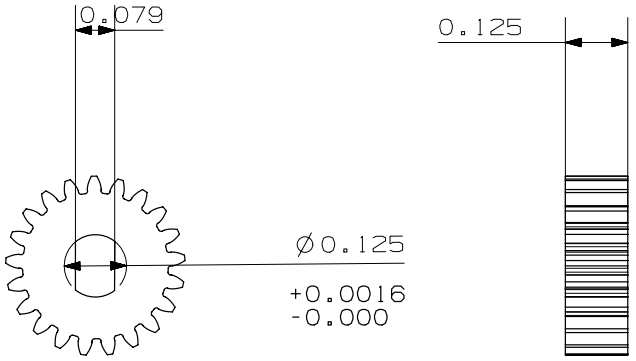




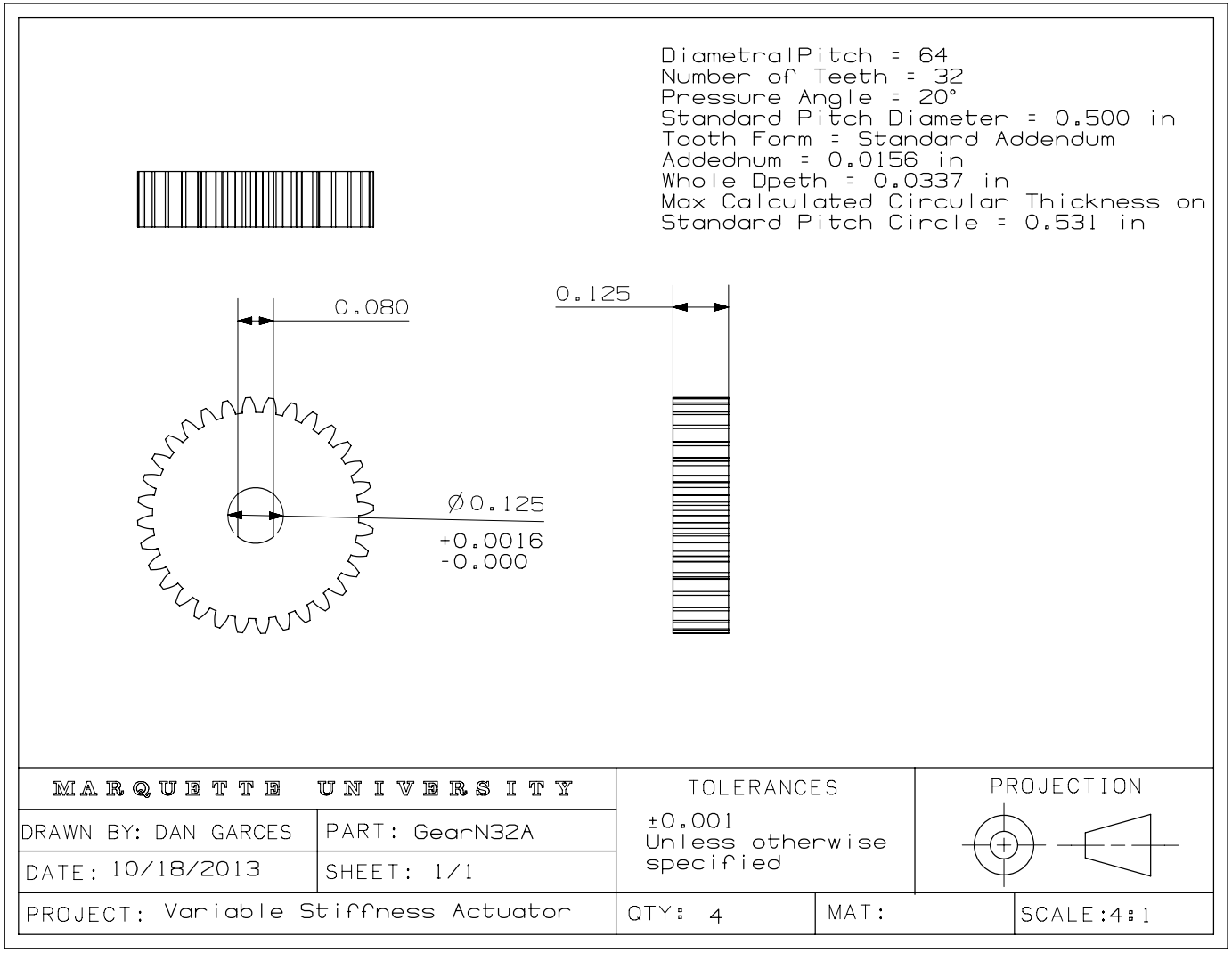
MARQUETTE UNIVERSITY		TOLERANCES	PROJECTION
DRAWN BY: DAN GARCES	PART: GearN18A	±0.001 Unless otherwise specified	
DATE: 10/18/2013	SHEET: 1/1		
PROJECT: Variable Stiffness Actuator		QTY: 2	MAT:	SCALE: 4:1

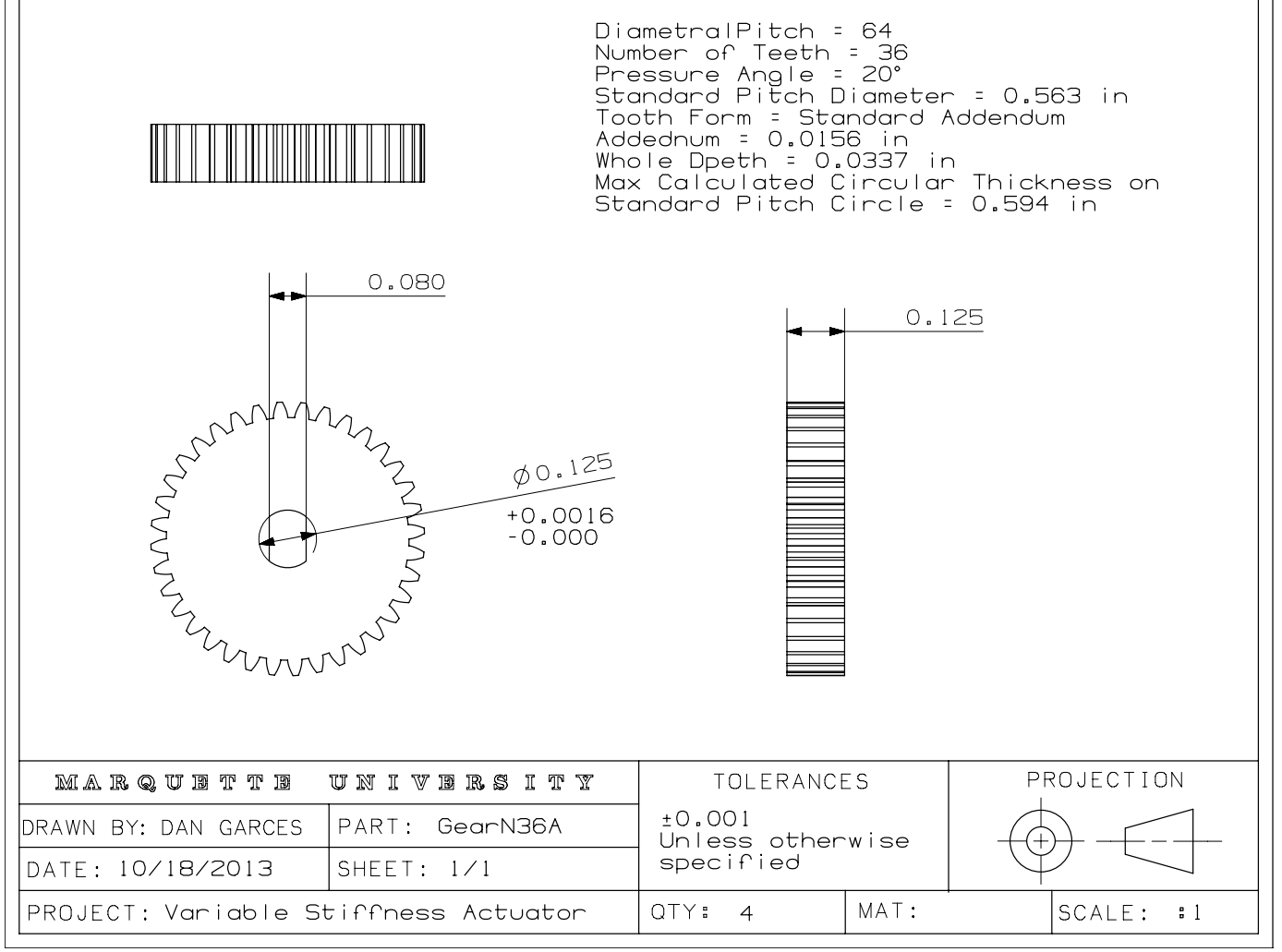


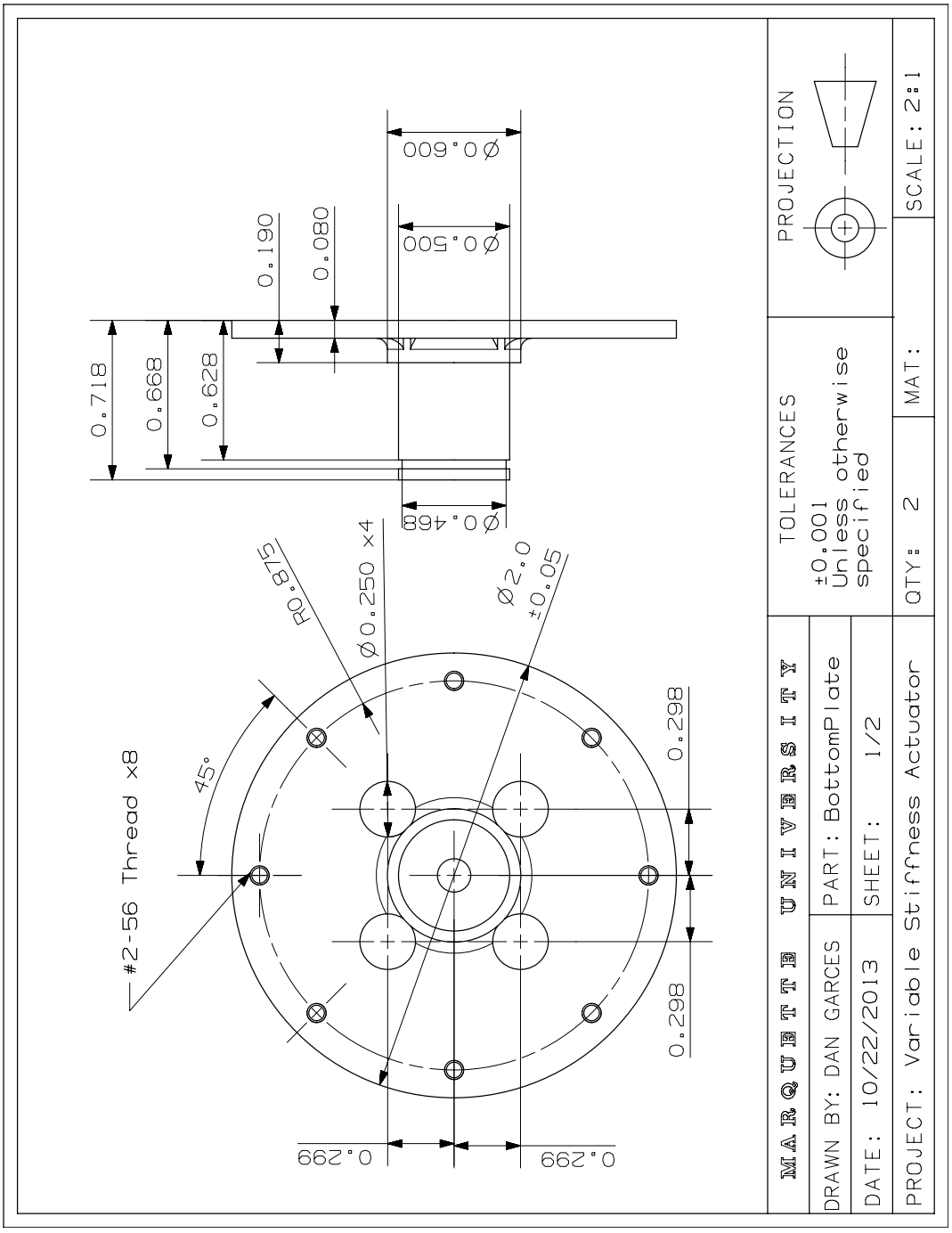
DiametralPitch = 64
Number of Teeth = 21
Pressure Angle = 20°
Standard Pitch Diameter = 0.328 in
Tooth Form = Standard Addendum
Addednum = 0.0156 in
Whole Dpeth = 0.0337 in
Max Calculated Circular Thickness on
Standard Pitch Circle = 0.359 in

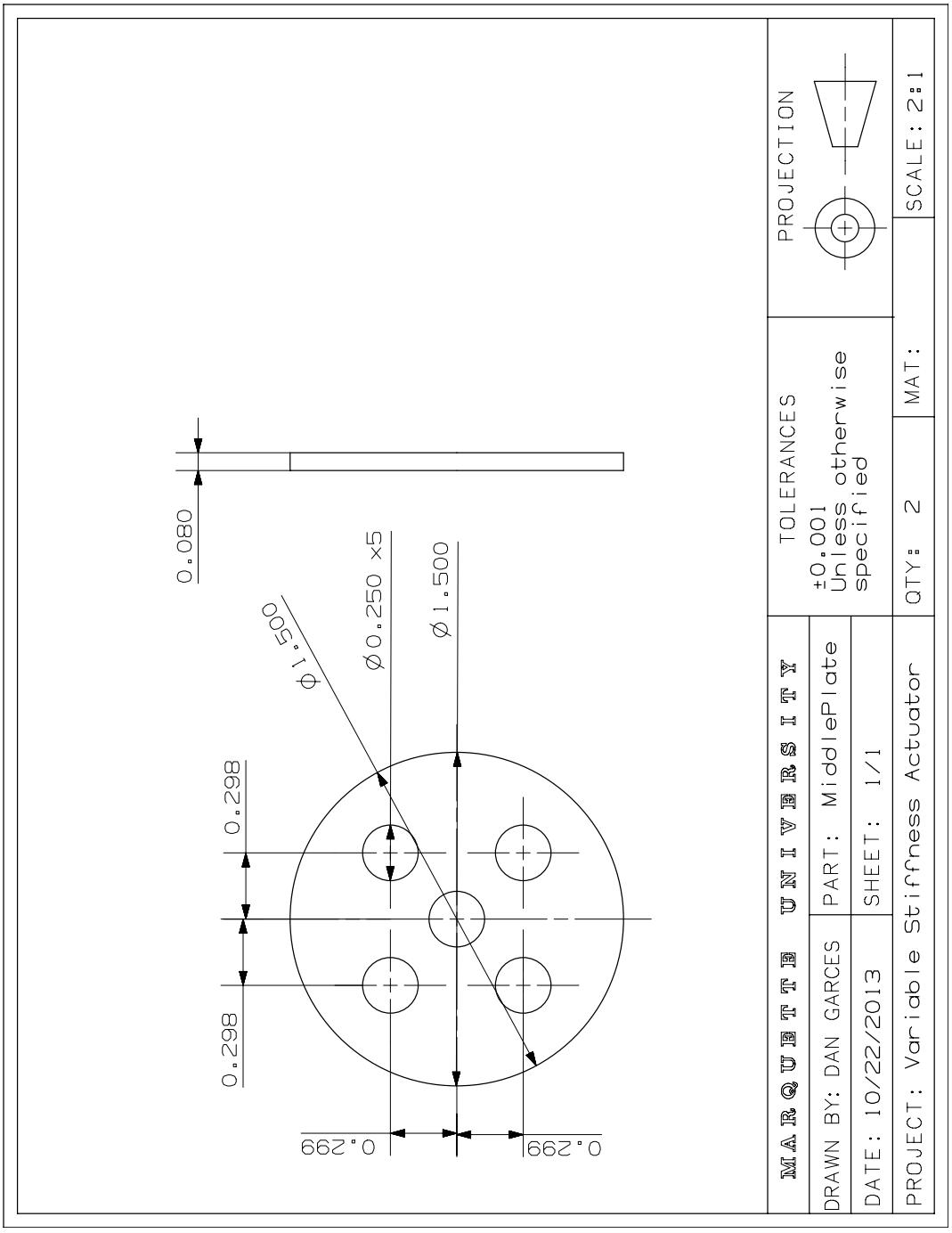


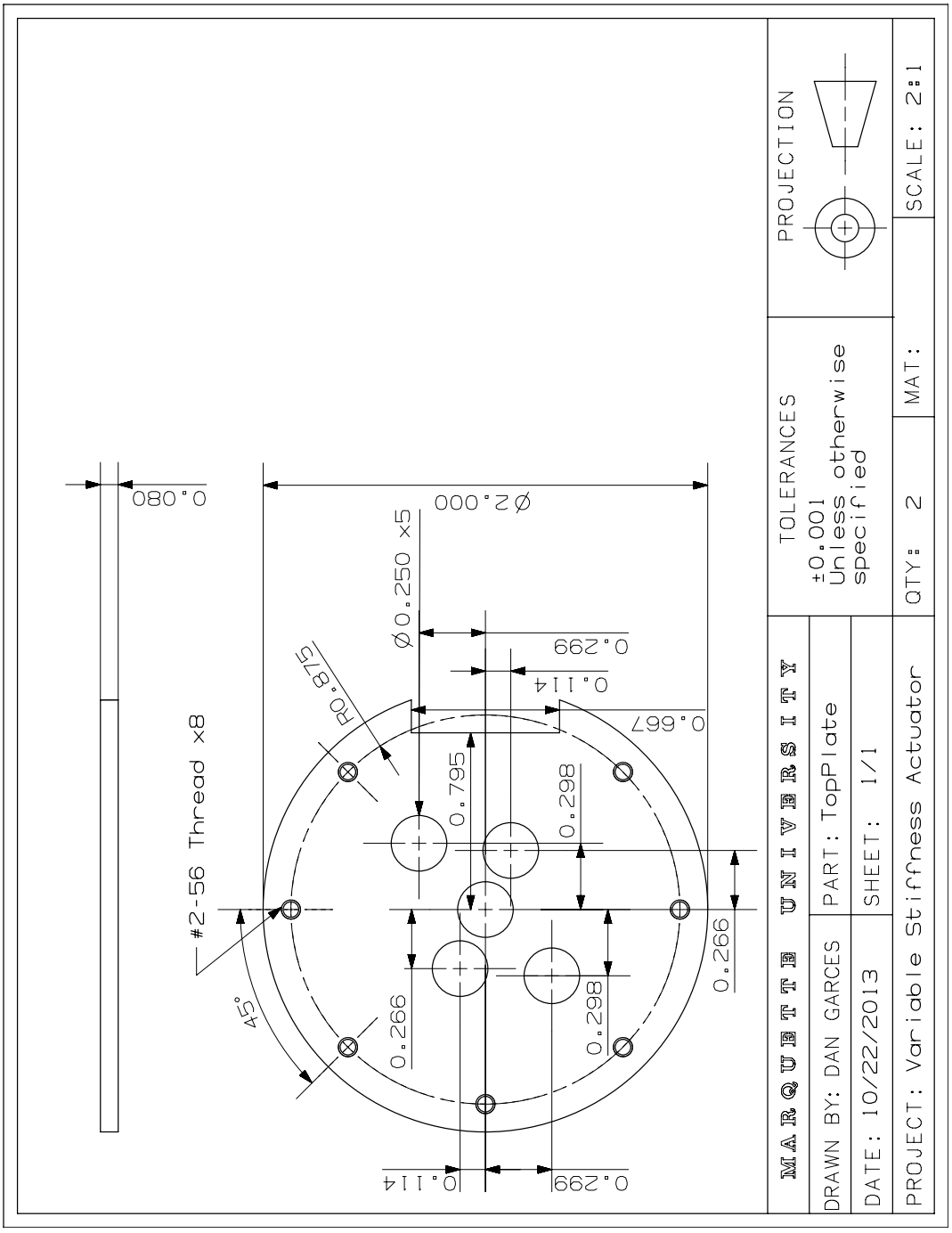
MARQUETTE UNIVERSITY		TOLERANCES	PROJECTION
DRAWN BY: DAN GARCES	PART: GearN21A	±0.001 Unless otherwise specified	
DATE: 10/18/2013	SHEET: 1/1		
PROJECT: Variable Stiffness Actuator		QTY: 4	MAT: SCALE: 4:1

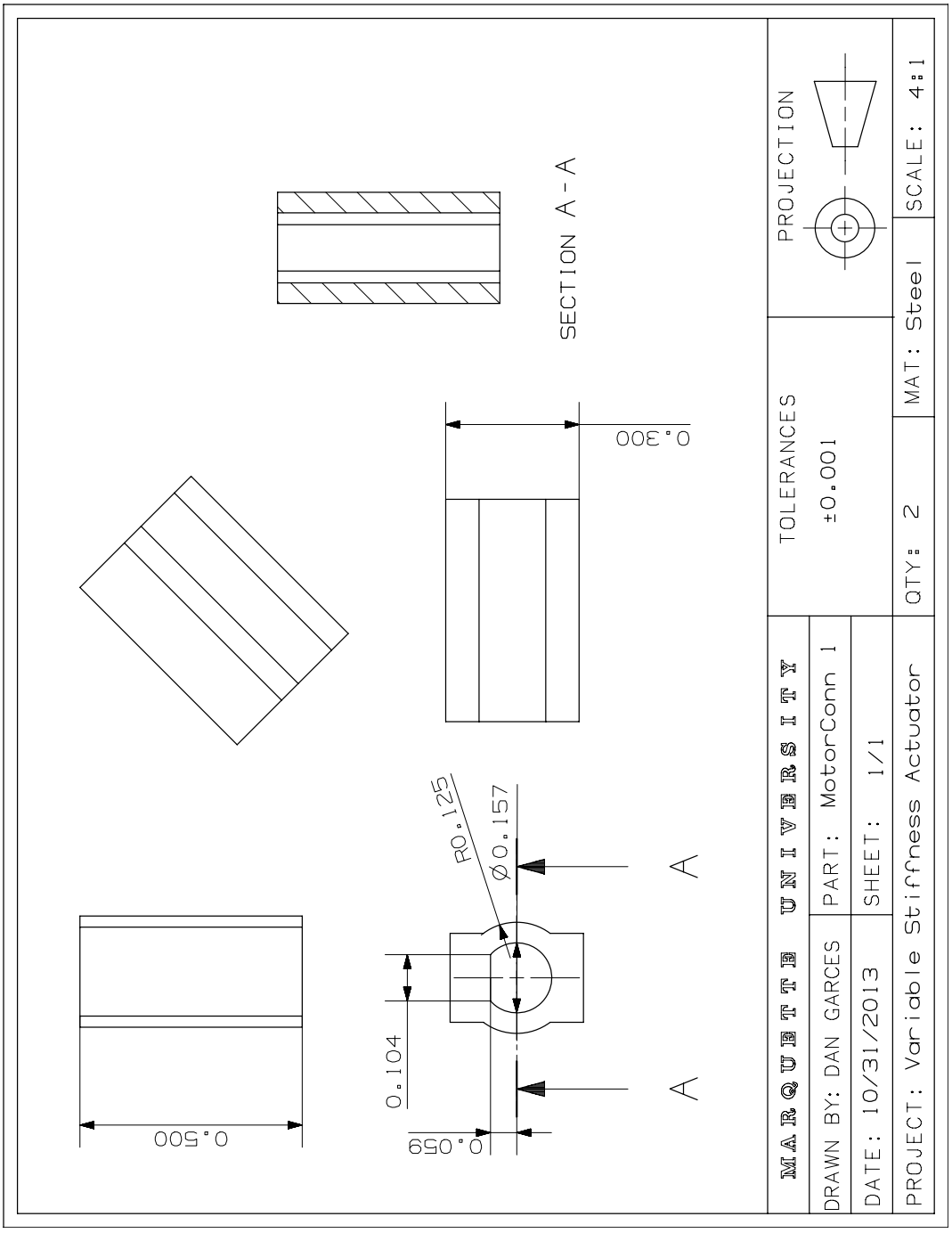

DiametralPitch = 64
Number of Teeth = 32
Pressure Angle = 20°
Standard Pitch Diameter = 0.500 in
Tooth Form = Standard Addendum
Addednum = 0.0156 in
Whole Dpeth = 0.0337 in
Max Calculated Circular Thickness on
Standard Pitch Circle = 0.531 in
0.080
0.125
Ø0.125
+0.0016
-0.000
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: GearN32A
DATE: 10/18/2013
SHEET: 1/1
TOLERANCES
±0.001
Unless otherwise
specified
PROJECTION
PROJECT: Variable Stiffness Actuator
QTY: 4
MAT:
SCALE:4:1

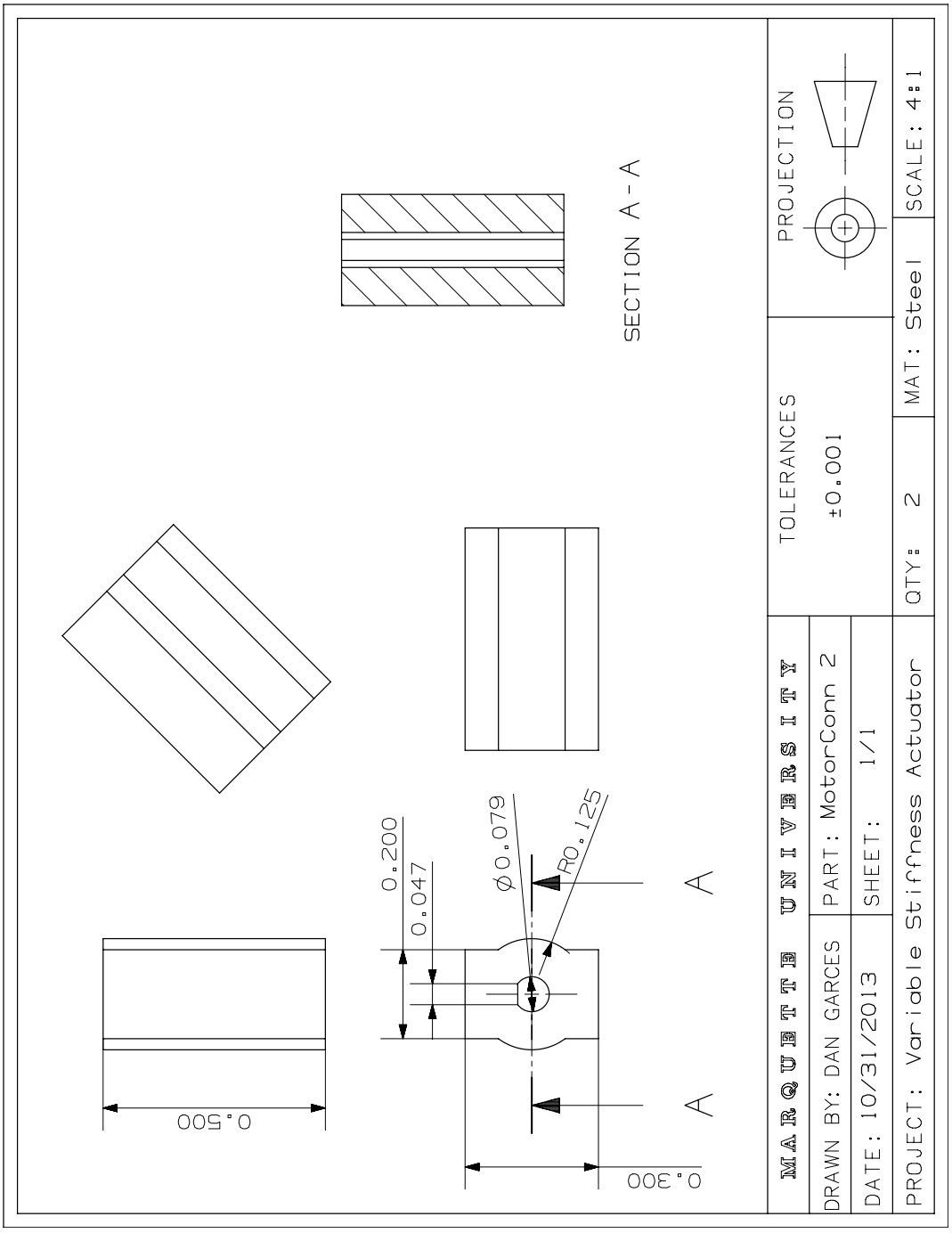

DiametralPitch = 64
Number of Teeth = 36
Pressure Angle = 20°
Standard Pitch Diameter = 0.563 in
Tooth Form = Standard Addendum
Addednum = 0.0156 in
Whole Dpeth = 0.0337 in
Max Calculated Circular Thickness on
Standard Pitch Circle = 0.594 in
0.080
Ø0.125
+0.0016
-0.000
0.125
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: GearN36A
DATE: 10/18/2013
SHEET: 1/1
PROJECT: Variable Stiffness Actuator
TOLERANCES
±0.001
Unless otherwise
specified
PROJECTION
QTY: 4
MAT:
SCALE: :1

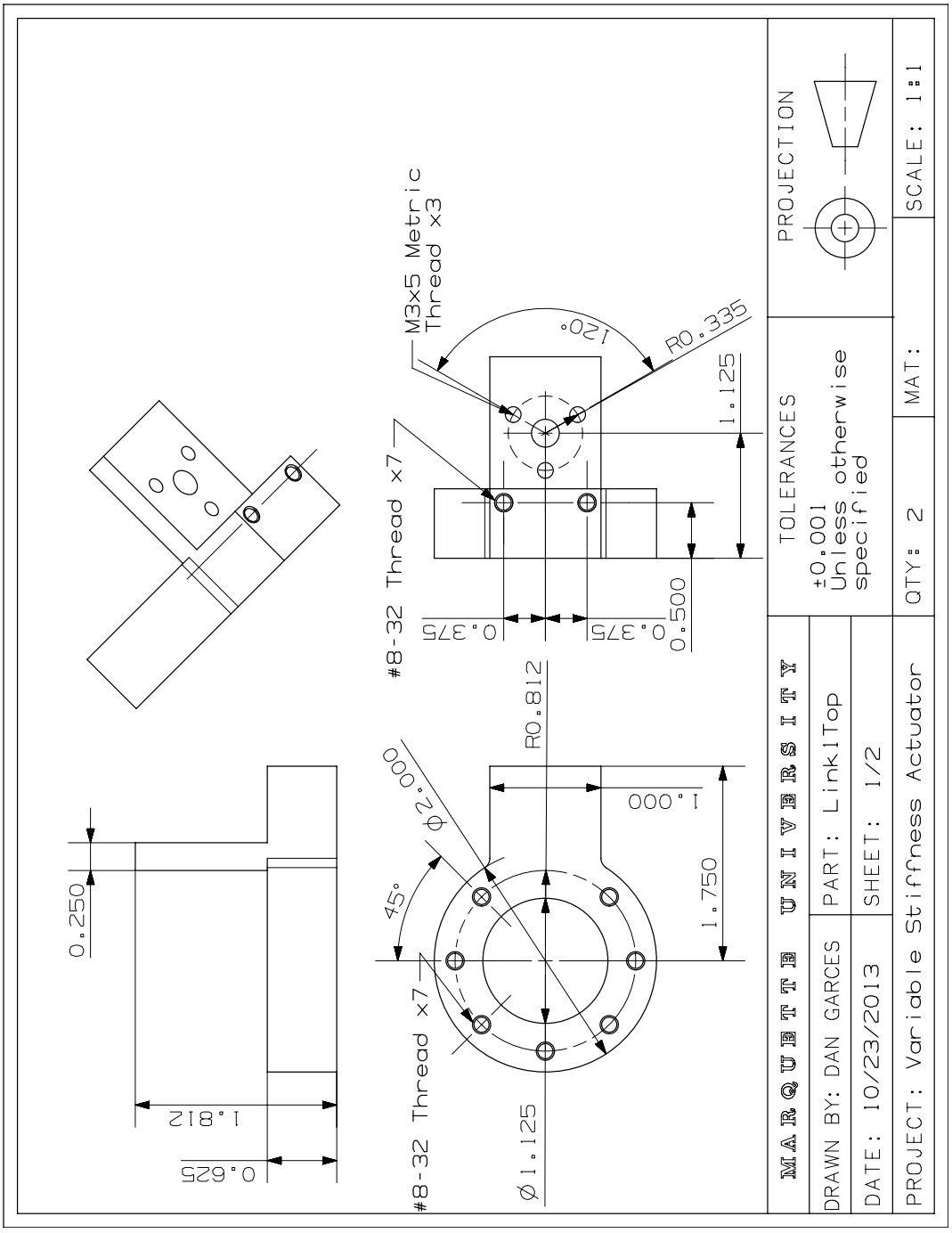

0.718
0.668
0.628
0.190
0.080
Ø0.600
Ø0.500
Ø0.468
#2-56 Thread ×8
45°
R0.875
Ø0.250 ×4
Ø2.0 ±0.05
0.298
0.298
0.299
0.299
TOLERANCES
±0.001
Unless otherwise specified
PROJECTION
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: BottomPlate
DATE: 10/22/2013
SHEET: 1/2
QTY: 2
MAT:
SCALE: 2:1
PROJECT: Variable Stiffness Actuator

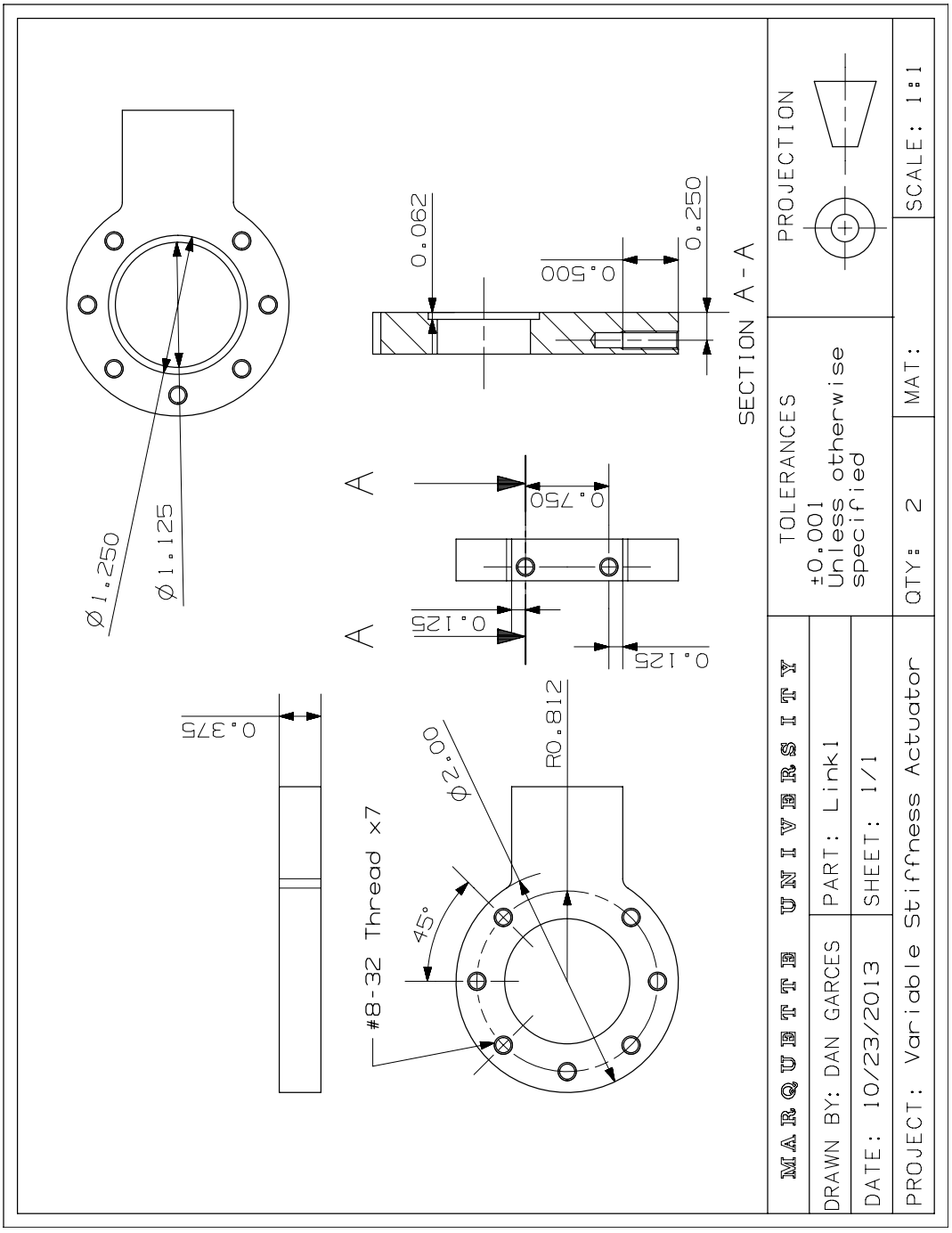

0.080
Ø1.500
Ø0.250 ×5
Ø1.500
0.298
0.298
0.299
0.299
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: MiddlePlate
DATE: 10/22/2013
SHEET: 1/1
PROJECT: Variable Stiffness Actuator
TOLERANCES
±0.001
Unless otherwise specified
QTY: 2
MAT:
PROJECTION
SCALE: 2:1

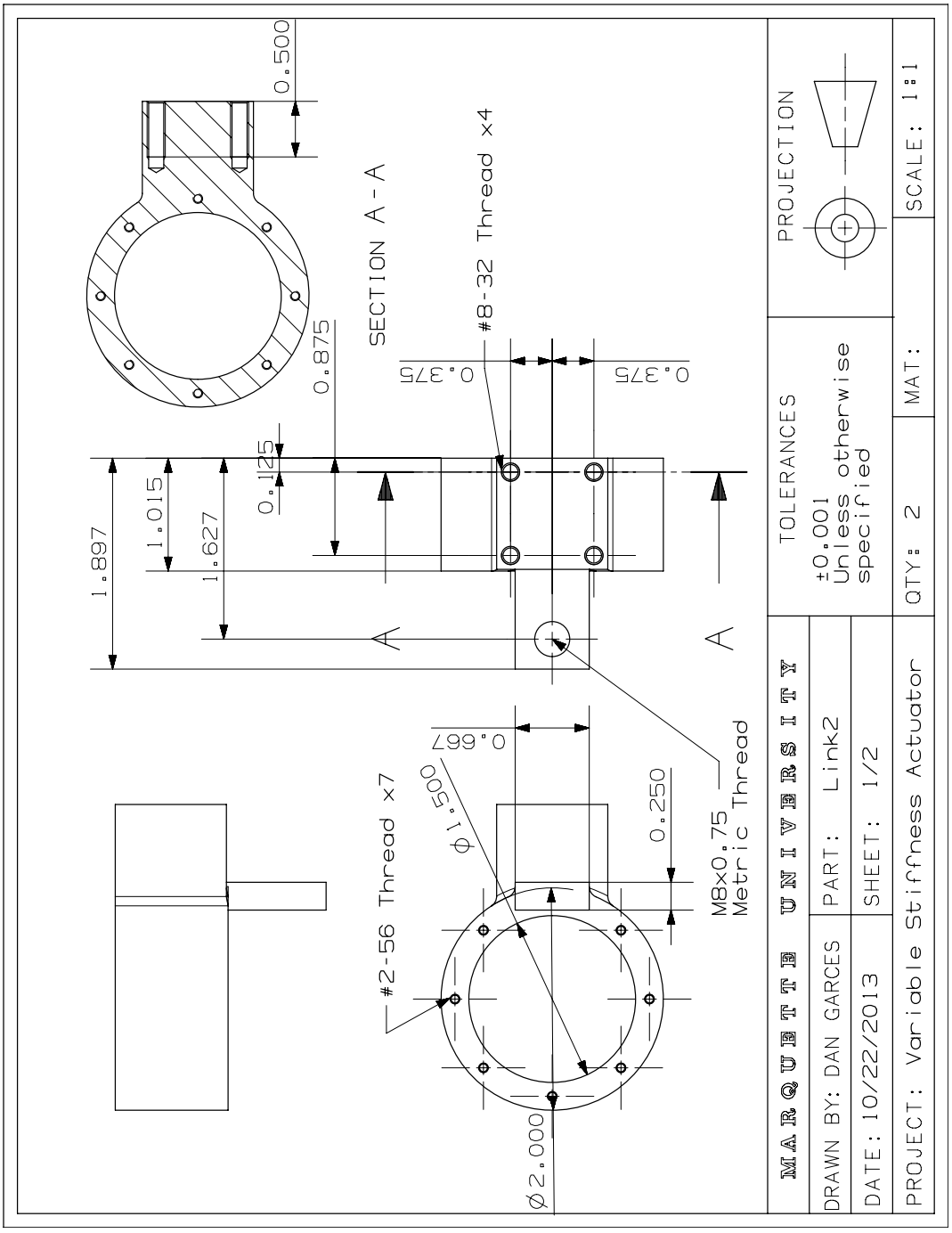

0.080
Ø2.000
Ø0.250 ×5
0.299
0.114
0.667
R0.875
0.795
0.298
#2-56 Thread ×8
45°
0.266
0.298
0.266
0.114
0.299
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: TopPlate
DATE: 10/22/2013
SHEET: 1/1
PROJECT: Variable Stiffness Actuator
TOLERANCES
±0.001
Unless otherwise specified
QTY: 2
MAT:
PROJECTION
SCALE: 2:1

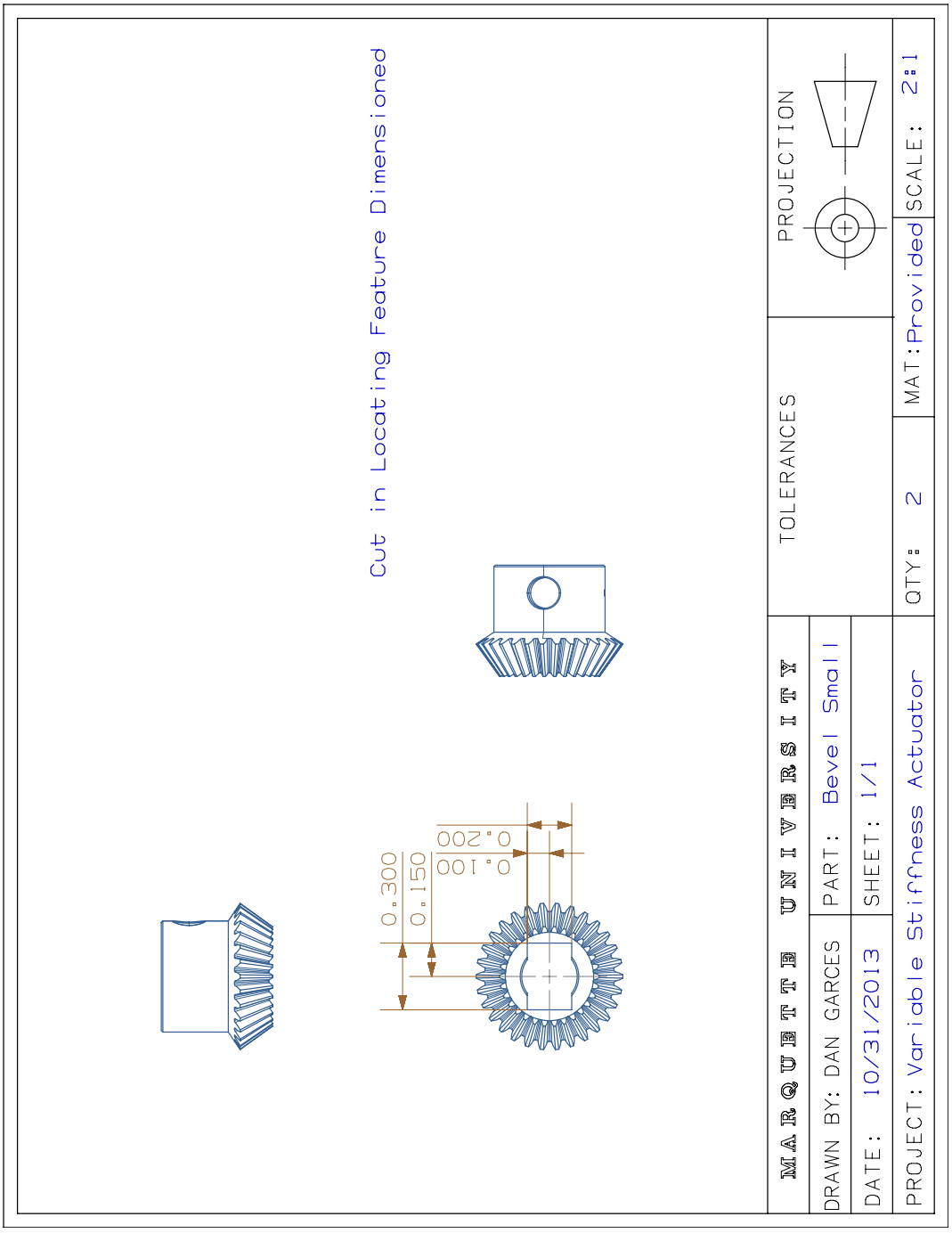

0.500
0.104
0.059
R0.125
Ø0.157
A
A
0.300
SECTION A-A
PROJECTION
TOLERANCES
±0.001
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: MotorConn 1
DATE: 10/31/2013
SHEET: 1/1
QTY: 2
MAT: Steel
SCALE: 4:1
PROJECT: Variable Stiffness Actuator

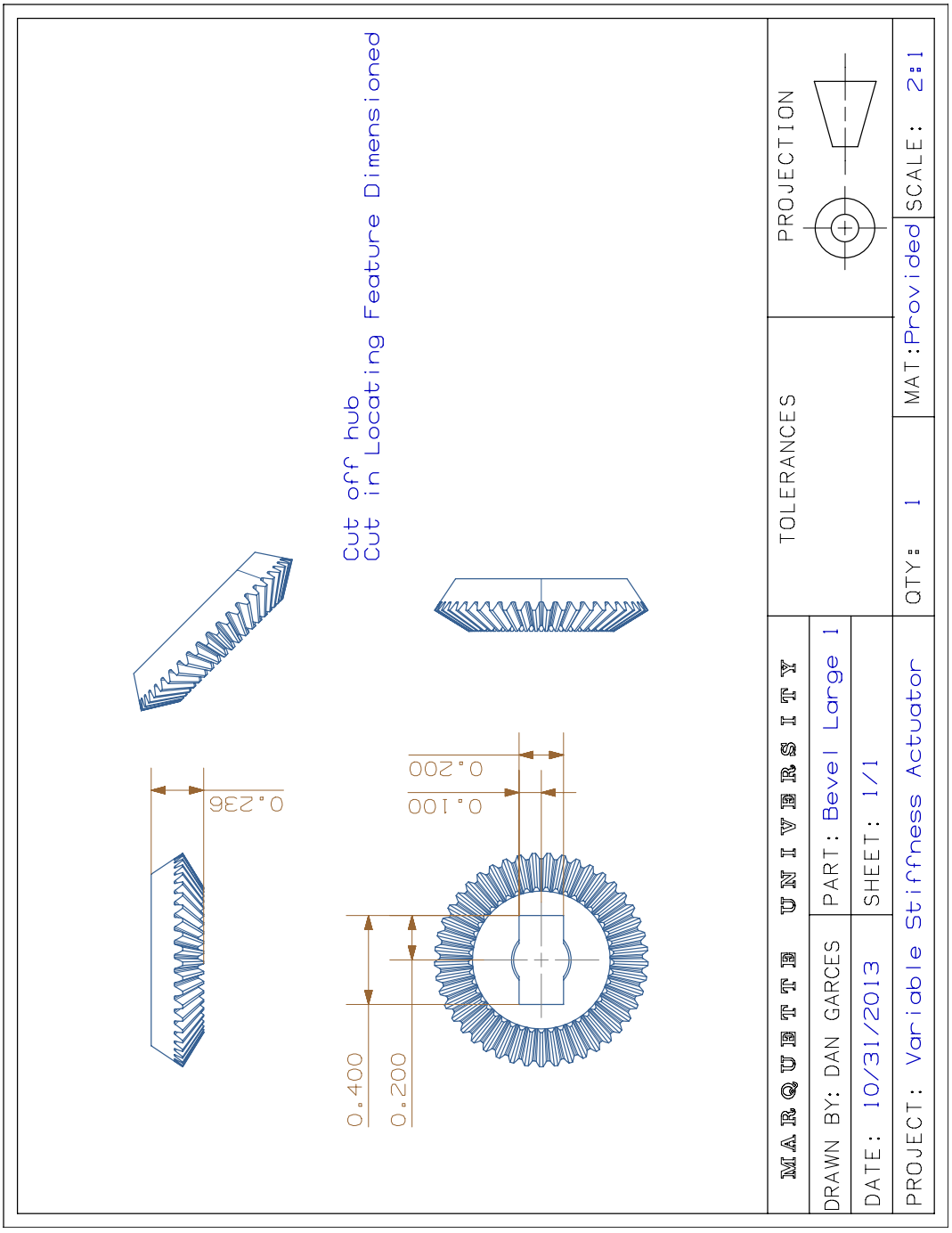

0.500
0.200
0.047
Ø0.079
R0.125
0.300
A
A
SECTION A - A
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: MotorConn 2
DATE: 10/31/2013
SHEET: 1/1
PROJECT: Variable Stiffness Actuator
TOLERANCES
±0.001
QTY: 2
MAT: Steel
PROJECTION
SCALE: 4:1

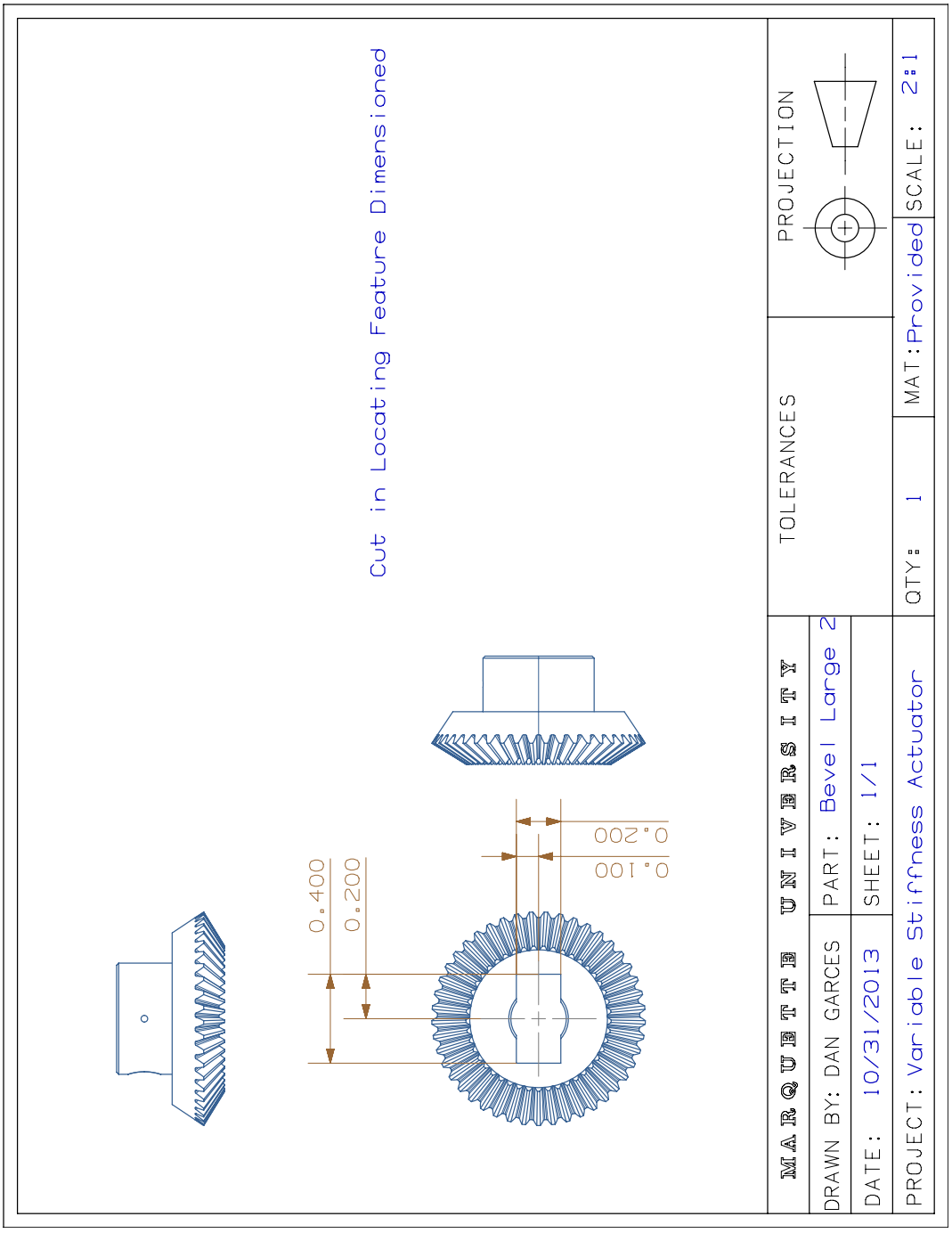

0.250
1.812
0.625
#8-32 Thread x7
Ø1.125
45°
Ø2.000
R0.812
1.000
1.750
M3x5 Metric
Thread x3
120°
R0.335
1.125
#8-32 Thread x7
0.375
0.375
0.500
PROJECTION
TOLERANCES
±0.001
Unless otherwise
specified
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: Link1ToP
DATE: 10/23/2013
SHEET: 1/2
QTY: 2
MAT:
SCALE: 1:1
PROJECT: Variable Stiffness Actuator


Ø1.250
Ø1.125
0.375
#8-32 Thread x7
45°
ϕ2.00
R0.812
A
A
0.750
0.125
0.125
0.062
0.500
0.250
SECTION A-A
PROJECTION
TOLERANCES
±0.001
Unless otherwise specified
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: Link1
DATE: 10/23/2013
SHEET: 1/1
QTY: 2
MAT:
SCALE: 1:1
PROJECT: Variable Stiffness Actuator


0.500
SECTION A - A
#8-32 Thread x4
0.875
0.375
0.375
0.125
1.015
1.627
1.897
A
A
M8x0.75
Metric Thread
#2-56 Thread x7
ϕ1.500
0.667
0.250
Ø2.000
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: Link2
DATE: 10/22/2013
SHEET: 1/2
PROJECT: Variable Stiffness Actuator
TOLERANCES
±0.001
Unless otherwise specified
PROJECTION
QTY: 2
MAT:
SCALE: 1:1


Cut in Locating Feature Dimensioned
0.300
0.150
Ø0.100
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: Bevel Small
DATE: 10/31/2013
SHEET: 1/1
TOLERANCES
PROJECTION
QTY: 2
MAT: Provided
SCALE: 2:1
PROJECT: Variable Stiffness Actuator


Cut off hub
Cut in Locating Feature Dimensioned
0.236
0.200
0.100
0.400
0.200
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: Bevel Large 1
DATE: 10/31/2013
SHEET: 1/1
TOLERANCES
PROJECTION
QTY: 1
MAT:Provided
SCALE: 2:1
PROJECT: Variable Stiffness Actuator


Cut in Locating Feature Dimensioned
0.400
0.200
0.100
0.200
MARQUETTE UNIVERSITY
DRAWN BY: DAN GARCES
PART: Bevel Large 2
DATE: 10/31/2013
SHEET: 1/1
PROJECT: Variable Stiffness Actuator
QTY: 1
MAT:Provided
SCALE: 2:1
TOLERANCES
PROJECTION